

**Department of Computer Science  
University of Delhi**

**Model Curricula under UGCF  
June 2022**

DRAFT

**TABLE-3**  
**Bachelor of (Field of Study/ Discipline) (Hons.)**

Semester	Core (DSC) File 1.	Elective (DSE) File 3	Generic Elective (GE)	Ability Enhancement Course (AEC)	Skill Enhancement Course (SEC)	Internship/ Apprentice- ship/Project/ Community outreach (2)	Value addition course (VAC)	Total Credits
	DSC - 1(4) Programming using Python		Choose one from a pool of courses GE-1 (4)	Choose one from a pool of AEC courses (2)	Choose one from a pool of courses (2)		Choose one from a pool of courses (2)	22 credits
	DSC - 2(4) Computer System Architecture							
	DSC - 3(4) Mathematics for Computing							
II	DSC- 4(4) Object Oriented Programming with C++		Choose one from a pool of courses GE-2 (4)	Choose one from a pool of AEC courses (2)	Choose one from a pool of courses (2)		Choose one from a pool of courses (2)	22 credits
	DSC- 5(4) Discrete Mathematical Structures							
	DSC- 6(4) Probability for Computing							
<b>Students on exit shall be awarded Undergraduate Certificate (in the field of Study/ Discipline) after securing the requisite 44 credits in Semesters I and II</b>								Total= 44
III	DSC - 7(4) Data Structures	Choose one from pool of courses, DSE- 1 (4)		Choose one from a pool of	Choose one SEC OR		Choose one from a pool of	22 credits

		OR Choose one from pool of courses, GE -3 (4)**	AEC courses (2)	Internship / Apprenticeship/ Project/ Com m unity Outreach (IAPC) (2) *	courses (2)	
	DSC- 8(4) Operating Systems	Data Analysis and Visualization (DAV)				
	DSC-9 (4) Numerical Optimization	OR Cryptography				
		OR Microprocessor				
IV	DSC - 10(4) Design and Analysis of Algorithms	Choose one from pool of courses, DSE- 2 (4) OR in the alternative choose one from pool of courses GE-4 (4)**	Choose one from a pool of AEC courses (2)	Choose one SEC <b>OR</b> ' Intern ship / Apprenticeship/ Project/ com m unity outreach (IAPC) (2) *	Choose one from a pool of courses (2)	22 Credits
	DSC - 11(4) Database Management System	Data Mining-I OR Combinatorial Optimisation				
	DSC - 12(4) Computer Networks	OR Data Privacy OR Introduction to Web Programming				
Students on exit shall be awarded Undergraduate Diploma (in the Field of Study/ Discipline) after securing the requisite 88 credits on completion of Semester IV						Total= 88
V	DSC - 13(4) Algorithms and Advanced Data Structures	Data Mining- II OR	Choose one from a pool of courses GE-5 (4)	Choose one SEC OR Intern ship / Apprenticeship / Project/ Com m unity Outreach (IAPC) (2) ***		22 credits
	DSC - 14(4) Theory of Computation	Network Security				

		OR					
	DSC- 15(4) Software Engineering	Unix Network Programming					

DRAFT

<b>VI</b>	DSC- 16(4) Artificial Intelligence	Choose one from a pool of courses DSE-4 (4) Social Network Analytics  OR Computer Graphics  OR Deep Learning  OR Ethical Hacking  OR Wireless Technology and Standards  OR Research Methodology (RM)	Choose one from a pool of courses GE-6 (4) "		Choose one SEC <b>OR</b> 'Int ern ship / Apprenticeship / Project / Research / Comm unity Outreach (2) * * *		22 credits
	DSC-17 (4) Machine Learning						
	DSC-18 (4) Introduction to Parallel Programming						

	<p align="center"><b>Students on exit shall be awarded Bachelor of (in the Field of Study/Discipline) Honours (3 years) after securing the requisite 132 credits on completion of Semester VI</b></p>						Total= 132
VII	DSC-19 (4) Compiler Design	<p>Choose three DSE (3x4) courses OR Choose two DSE- (2x4) and one GE (4) " course OR Choose one DSE (4) and two GE (2x4) courses (total = 12)#</p> <ul style="list-style-type: none"> <li>• Digital Image Processing</li> <li>• Advanced Algorithms</li> <li>• Reinforcement Learning</li> <li>• Cyber Forensic</li> <li>• Advanced TCP/IP</li> <li>• Mobile and Satellite Communication</li> <li>• Research Methodology (RM)</li> </ul>				<p>Dissertation on Major (6) OR Dissertation on Minor (6) OR Academic project/ Entrepreneurship (6)</p>	22 credits
VIII	DSC -20 (4) Information Security	<p>Choose three DSE (3x4) courses OR Choose two DSE- (2x4) and one GE (4) course OR Choose one DSE (4) and two GE (2x4) courses (total = 12)#</p> <ul style="list-style-type: none"> <li>• Information and Image Retrieval</li> <li>• Natural Language Processing</li> <li>• IT Infrastructure</li> </ul>				<p>Dissertation on Major (6) OR Dissertation on Minor (6) OR Academic project/ Entrepreneurship (6)</p>	22 credits

		<p>Management</p> <ul style="list-style-type: none"> <li>• Critical Infrastructure Management</li> <li>• Blockchain and its applications</li> <li>• Cloud Computing</li> <li>• IOT</li> <li>• Distributed Algorithms</li> </ul>					
<p><b>Students on exit shall be awarded Bachelor of (in the Field of Study/Discipline) (Honours with Research/Academic Projects/Entrepreneurship) or (Honours with Research in Discipline- I (Major) with Discipline-2 (Minor) after securing the requisite 176 credits on completion of Semester VIII</b></p>							Total = 176

DRAFT

# Table of Contents

1. [Preamble](#)
2. [Discipline Specific Graduate Attributes](#)
3. [Table of Core Courses](#)
4. [Syllabi of Core Courses](#)

## 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 (generic 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.

## **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	Desirable but not Prerequisites	DSGA Contributed to
I	<a href="#">DSC 01</a>	Programming using Python	3	0	1	4	Nil	DSGA1
I	<a href="#">DSC 02</a>	Computer System Architecture	3	0	1	4	Nil	DSGA3
I	<a href="#">DSC 03</a>	Mathematics for computing	3	0	1	4	Nil	DSGA4
II	<a href="#">DSC 04</a>	Object Oriented Programming with C++	3	0	1	4	Nil	DSGA1
II	<a href="#">DSC 05</a>	Discrete Mathematical Structures	3	0	1	4	Nil	DSGA4
II	<a href="#">DSC 06</a>	Probability for Computing	3	0	1	4	Nil	DSGA4
III	<a href="#">DSC 07</a>	Data Structures	3	0	1	4	<a href="#">DSC 04/</a> a course in C/C++ at plus 2 level/**	DSGA1, DSGA2
III	<a href="#">DSC 08</a>	Operating	3	0	1	4	<a href="#">DSC</a>	DSGA3

		Systems					<a href="#">04/a</a> course in C/C++ at plus 2 level/**	
III	<a href="#">DSC 09</a>	Numerical Optimization	3	0	1	4	Nil	DSGA4
IV	<a href="#">DSC10</a>	Design and Analysis of Algorithms	3	0	1	4	<a href="#">DSC07</a>	DSGA1, DSGA2
IV	<a href="#">DSC 11</a>	Database Management System	3	0	1	4	<a href="#">DSC01/</a> a course in Python at plus 2 level/** , <a href="#">DSC08</a>	DSGA3
IV	<a href="#">DSC 12</a>	Computer Networks	3	0	1	4	<a href="#">DSC04/a</a> course in C/C++ at plus 2 level/** , <a href="#">DSC 07</a> , <a href="#">DSC 08</a>	DSGA3
V	<a href="#">DSC 13</a>	Algorithms and Advanced Data Structures	3	0	1	4	<a href="#">DSC07</a> , <a href="#">DSC10</a>	DSGA1, DSGA2.
V	<a href="#">DSC 14</a>	Theory of Computation	3	1	0	4	<a href="#">DSC04/a</a> course in C/C++ at plus 2 level/**  <a href="#">DSC05</a>	DSGA4

V	<a href="#">DSC 15</a>	Software Engineering	3	0	1	4	<a href="#">DSC01</a> ,/ <a href="#">DSC04</a> ,/ a course in C/C++/P ython at plus 2 level/**	DSGA5
VI	<a href="#">DSC 16</a>	Artificial Intelligence	3	0	1	4	<a href="#">DSC 01</a> <a href="#">DSC03</a> <a href="#">DSC 06</a> <a href="#">DSC 07</a> <a href="#">DSC 09</a> <a href="#">DSC10</a> <a href="#">DSC14</a>	DSGA6
VI	<a href="#">DSC 17</a>	Machine Learning	3	0	1	4	<a href="#">DSC 01</a> <a href="#">DSC03</a> <a href="#">DSC 06</a> <a href="#">DSC 07</a> <a href="#">DSC 09</a> <a href="#">DSC10</a>	DSGA6
VI	<a href="#">DSC 18</a>	Introduction to Parallel Programming	3	0	1	4	<a href="#">DCS 02</a> , <a href="#">DSC04</a> , <a href="#">DSC 07</a> , <a href="#">DSC 08</a>	DSGA7
VII	<a href="#">DSC 19</a>	Compiler Design	3	0	1	4	<a href="#">DSC-14</a> :	DSGA8
VIII	<a href="#">DSC 20</a>	Information Security	3	0	1	4	<a href="#">DSC 01</a> <a href="#">DSC04</a> , <a href="#">DSC 07</a> <a href="#">DSC 08</a> <a href="#">DSC 11</a> <a href="#">DSC 12</a>	DSGA1, DSGA9

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**

#### **Course Objective**

The course is designed to introduce programming concepts using Python to students. The course aims to develop structured as well as object-oriented programming skills using Python. The course also aims to achieve competence amongst its students to develop correct and efficient Python programs to solve real life problems.

#### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Develop, document, and debug modular Python programs of reasonable complexity.
2. Implement arrays and user defined functions in Python.
3. Write programs in Python using dynamic memory allocation, handling external files, interrupts and exceptions.

4. Solve real life problems of reasonable complexity using suitable and efficient programming constructs in Python.
5. Solve real life problems of reasonable complexity using the concepts of object oriented programming in Python.

## Syllabus

**Unit 1 Introduction to Programming:** Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

**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.

**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.

**Unit 4 Object Oriented Programming:** Introduction to classes, objects and methods; Standard libraries.

**Unit 5 File and exception handling:** File handling through libraries; Errors and exception handling.

## References

1. Taneja, S., Kumar, N. *Python Programming- A modular Approach*, 1<sup>st</sup> edition, Pearson Education India, 2018.
2. Balaguruswamy E. *Introduction to Computing and Problem Solving using Python*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.

## Additional References

- (i) Brown, Martin C. *Python: The Complete Reference*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.
- (ii) Guttag, J.V. *Introduction to computation and programming using Python*, 2<sup>nd</sup> edition, MIT

Press, 2016.

### Suggested Practical List

1. WAP to find the roots of a quadratic equation
2. WAP to accept a number 'n' and
  - a. Check if 'n' is prime
  - b. Generate all prime numbers till 'n'
  - c. Generate first 'n' prime numbers

This program may be done using functions

3. WAP to create a pyramid of the character '\*' and a reverse pyramid

```
*
***
*****
*****
*****
```

```
*****
*****
*****
***
*
```

4. WAP that accepts a character and performs the following:
  - a. print whether the character is a letter or numeric digit or a special character
  - b. if the character is a letter, print whether the letter is uppercase or lowercase
  - c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. WAP to perform the following operations on a string
  - a. Find the frequency of a character in a string.
  - b. Replace a character by another character in a string.
  - c. Remove the first occurrence of a character from a string.
  - d. Remove all occurrences of a character from a string.
6. WAP to swap the first n characters of two strings.
7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.

8. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:
  - a. 'for' loop
  - b. list comprehension
  
9. WAP to read a file and
  - a. Print the total number of characters, words and lines in the file.
  - b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
  - c. Print the words in reverse order.
  - d. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.
  
10. WAP to define a class Point with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method distance to calculate the distance between any two point objects.
  
11. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
  
12. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:
  - a) Print half the values of the tuple in one line and the other half in the next line.
  - b) Print another tuple whose values are even numbers in the given tuple.
  - c) Concatenate a tuple t2=(11,13,15) with t1.
  - d) Return maximum and minimum value from this tuple
  
13. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

## **DSC 02: Computer System Architecture**

### **Course Objective**

This course introduces the students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized



together to architect a digital computer system.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Design and Simplify Combinational and sequential circuits using basic building blocks.
2. Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
3. **Explain** instruction cycle, pipelining and interrupts.
4. **Explain** data communication between CPU, memory and I/O devices.
5. Simulate the design of a basic computer using a software tool.

### **Syllabus**

**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.

**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.

**Unit 3 Data Representation and Basic Computer Arithmetic:** Number System,  $r$  and  $(r-1)$ 's Complements, data representation and arithmetic operations.

**Unit 4 Basic Computer Organization and Design:** Bus organization, Micro programmed vs Hardwired Control , Instruction Codes, Instruction Format, Instruction Cycle, Instruction pipelining, Memory Reference, Register Reference and Input Output Instructions, Program Interrupt and Interrupt Cycle.

**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)

**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.

## References

1. David A. Patterson and John L. Hennessy. “*Computer Organization and Design : The Hardware/Software interface*”, 5<sup>th</sup> edition, Elsevier, 2012.
2. Mano, M. *Computer System Architecture*, 3<sup>rd</sup> edition, Pearson Education, 1993.

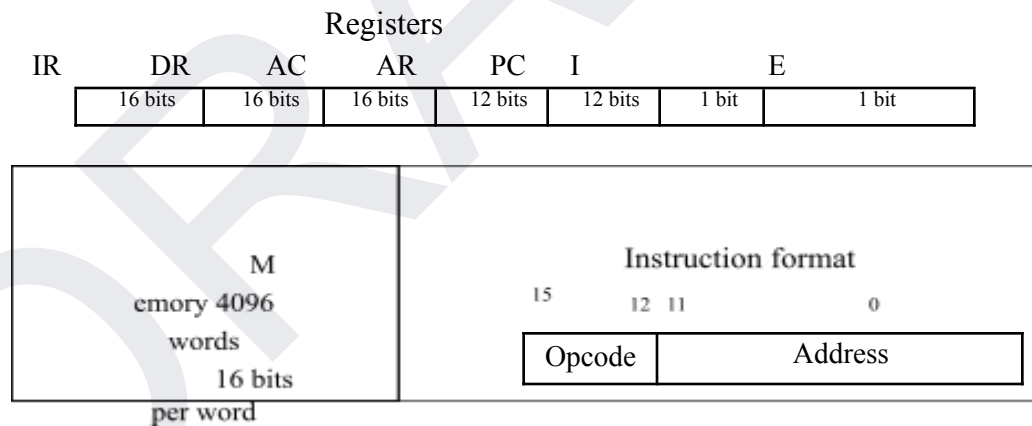
## Additional References

- (i) Mano, M. *Digital Design*, Pearson Education Asia, 1995.
- (ii) Null, L., & Lobur, J. *The Essentials of Computer Organization and Architecture*.  
5<sup>th</sup> edition. (Reprint) Jones and Bartlett Learning, 2018.
- (iii) Stallings, W. *Computer Organization and Architecture Designing for Performance*  
8<sup>th</sup> edition, Prentice Hall of India, 2010.

## Suggested Practical List

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture:



## Basic Computer Instructions

Memory Reference		Register Reference	
Symbol	Hex	Symbol	Hex
AND	0xxx	CLA	7800
ADD	1xxx	CLE	7400
LDA	2xxx	CMA	7200
STA	3xxx	CME	7100
BUN	4xxx	CIR	7080

BSA	5xxx	Addressing	CIL	7040
ISZ	6xxx		INC	7020
AND_I	8xxx	Indirect Addressing	SPA	7010
ADD_I	9xxx		SNA	7008
LDA_I	Axxx		SZA	7004
STA_I	Bxxx		SZE	7002
BUN_I	Cxxx		HLT	7001
BSA_I	Dxxx		INP	F800
ISZ_I	Exxx		OUT	F400

DRAFT

Refer to Chapter-5 of reference 1 for description of instructions.

Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

2. Create a Fetch routine of the instruction cycle.
3. Write an assembly program to simulate ADD operation on two user-entered numbers.
4. Write an assembly program to simulate SUBTRACT operation on two user-entered numbers.
5. Write an assembly program to simulate the following logical operations on two user-entered numbers.
  - i. AND
  - ii. OR
  - iii. NOT
  - iv. XOR
  - v. NOR
  - vi. NAND
6. Write an assembly program for simulating following memory-reference instructions.
  - i. ADD
  - ii. LDA
  - iii. STA
  - iv. BUN
  - v. ISZ
7. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
  - i. CLA
  - ii. CMA
  - iii. CME
  - iv. HLT

8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
  - i. INC
  - ii. SPA
  - iii. SNA
  - iv. SZE
  
9. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
  - i. CIR
  - ii. CIL
  
10. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
  
11. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.

### **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. Strang Gilbert. *Introduction to Linear Algebra*, 5<sup>th</sup> Edition, Wellesley-Cambridge Press, 2021.
2. Kreyszig Erwin. *Advanced Engineering Mathematics*, 10<sup>th</sup> Edition, Wiley, 2015.
3. Strang Gilbert. *Linear Algebra and Learning from Data*, 1<sup>st</sup> Edition, Wellesley-Cambridge Press, 2019.

4. Jain R. K., Iyengar S.R. K. *Advanced Engineering Mathematics*, 5<sup>th</sup> Edition, Narosa, 2016.

### **Additional References**

- (i) Deisenroth, Marc Peter, Faisal A. Aldo and Ong Cheng Soon. *Mathematics for Machine Learning*, 1<sup>st</sup> Edition, Cambridge University Press, 2020.
- (ii) Lipschutz Seymour and Lipson Marc. *Schaum's Outline of Linear Algebra*, 6<sup>th</sup> Edition, McGraw Hill, 2017.

### **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.
1. 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  $\mathbb{R}^n$ / 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++**

### **Course Objective**

This course is designed to introduce programming concepts using C++ to students. The course aims to develop structured as well as object-oriented programming skills using C++ programming language. The course also aims to achieve competence amongst its students to develop correct and efficient C++ programs to solve problems spanning multiple domains.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Write simple programs using built-in data types of C++.
2. Implement arrays and user defined functions in C++.
3. Write programs using dynamic memory allocation, handling external files, interrupts and exceptions.
4. Solve problems spanning multiple domains using suitable programming constructs in C++.
5. Solve problems spanning multiple domains using the concepts of object oriented programming in C++.

### **Syllabus**

**Unit 1 Introduction to C++:** Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++.

**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

**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

**Unit 4 Pointers and References:** Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references



**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

## References

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

## Additional References

- (i) Herbert Schildt, C++: *The Complete Reference*, 4<sup>th</sup> Edition, McGraw Hill, 2003
- (ii) A. B. Forouzan, Richard F. Gilberg, *Computer Science: A Structured Approach using C++*, 2<sup>nd</sup> edition, Cengage Learning, 2010

## Suggested Practical list

1. Write a program to compute the sum of the first n terms of the following series:

$$S = 1 - \frac{1}{2^2} + \frac{1}{3^3} - \frac{1}{4^4} + \frac{1}{5^5} - \dots$$

The number of terms n is to be taken from the user through the command line. If the command line argument is not found then prompt the user to enter the value of n.

2. Write a program to remove the duplicates from an array.
3. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
4. Write a menu driven program to perform string manipulation (without using inbuilt string functions):
  - a. Show address of each character in string
  - b. Concatenate two strings.
  - c. Compare two strings
  - d. Calculate length of the string (use pointers)
  - e. Convert all lowercase characters to uppercase
  - f. Reverse the string
  - g. Insert a string in another string at a user specified position
5. Write a program to merge two ordered arrays to get a single ordered array.

6. Write a program to search a given element in a set of N numbers using Binary search  
(i) with recursion (ii) without recursion.
7. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.
8. Create a Matrix class. Write a menu-driven program to perform following Matrix operations (exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by the main() function):
  - a. Sum
  - b. Product
  - c. Transpose
9. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism.
10. Create a Triangle class. Add exception handling statements to ensure the following conditions: all sides are greater than 0 and sum of any two sides are greater than the third side. The class should also have overloaded functions for calculating the area of a right angled triangle as well as using Heron's formula to calculate the area of any type of triangle.
11. Create a class Student containing fields for Roll No., Name, Class, Year and Total Marks. Write a program to store 5 objects of Student class in a file. Retrieve these records from the file and display them.
12. Copy the contents of one text file to another file, after removing all whitespaces.

### **DSC 05: Discrete Mathematical Structures**

#### **Course Objective**

This course is designed as a foundational course to make students learn about the mathematical constructs that are used in Computer Science such as Boolean algebra, sets, relations, functions, principles of counting, and recurrences. In this course, the knowledge of mathematical notation, ideas and concepts learnt at the pre-college levels is extended to orient the students towards mathematical thinking required in Computer Science.

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Relate mathematical concepts and terminology to examples in the domain of Computer Science.
2. Model real world problems using various mathematical constructs.
3. Use different proofing techniques; construct simple mathematical proofs using logical arguments.
4. Formulate mathematical claims and construct counterexamples.

## Syllabus

**Unit 1 Sets, Functions, Sequences and Summations, Relations:** Sets: Set Operations, Computer Representation of Sets, Countable and Uncountable Sets, 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.

**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.

**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.

**Unit 4 Combinatorics/Counting:** The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations.

**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.

**Unit 6 Recurrence:** Recurrence Relations, Generating Functions, Linear Recurrence Relations with Constant Coefficients and their solution.

## References

1. Liu, C.L., Mohapatra, D.P. *Elements of Discrete Mathematics: A Computer Oriented Approach*, 4<sup>th</sup> edition, Tata McGraw Hill, 2017.
2. Rosen, K.H.. *Discrete Mathematics and Its Applications*, 8<sup>th</sup> edition, Mc Graw Hill, 2018.

### **Additional References**

- (i) Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4<sup>th</sup> edition, Prentice Hall of India. 2022.
- (ii) Trembley, J.P., Manohar, R. *Discrete Mathematical Structures with Application to Computer Science*, Tata McGraw Hill, 1997.
- (iii) Albertson, M. O. and Hutchinson, J. P. *Discrete Mathematics with Algorithms*, John Wiley and Sons, 1988.

### **Suggested Practical list**

1. Create a class SET. Create member functions to perform the following SET operations:
  - 1) ismember: check whether an element belongs to the set or not and return value as true/false.
  - 2) powerset: list all the elements of the power set of a set .
  - 3) subset: Check whether one set is a subset of the other or not.
  - 4) union and Intersection of two Sets.
  - 5) complement: Assume Universal Set as per the input elements from the user.
  - 6) set Difference and Symmetric Difference between two sets.
  - 7) cartesian Product of Sets.

Write a menu driven program to perform the above functions on an instance of the SET class.

2. Create a class RELATION, use Matrix notation to represent a relation. Include member functions to check if the relation is Reflexive, Symmetric, Anti-symmetric, Transitive. Using these functions check whether the given relation is: Equivalence or Partial Order relation or None
3. Write a Program that generates all the permutations of a given set of digits, with or without repetition.
4. For any number n, write a program to list all the solutions of the equation  $x_1 + x_2 + x_3 + \dots + x_n = C$ , where C is a constant ( $C \leq 10$ ) and  $x_1, x_2, x_3, \dots, x_n$  are nonnegative integers, using brute force strategy.

5. Write a Program to evaluate a polynomial function. (For example store  $f(x) = 4n^2 + 2n + 9$  in an array and for a given value of  $n$ , say  $n = 5$ , compute the value of  $f(n)$ ).
6. Write a Program to check if a given graph is a complete graph. Represent the graph using the Adjacency Matrix representation.
7. Write a Program to check if a given graph is a complete graph. Represent the graph using the Adjacency List representation.
8. Write a Program to accept a directed graph  $G$  and compute the in-degree and out-degree of each vertex.

### **DSC 06: Probability for Computing**

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.

#### **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. Ross Sheldon M. *Introduction to Probability Models*, 12<sup>th</sup> Edition, Elsevier, 2019.
2. Trivedi, K.S. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, 2<sup>nd</sup> edition, Wiley, 2015.
3. Deisenroth, Marc Peter, Faisal A. Aldo and Ong Cheng Soon, *Mathematics for Machine Learning*, 1<sup>st</sup> edition, Cambridge University Press, 2020.
4. Ian F. Blake, *An Introduction to Applied Probability*, John Wiley.

## Additional References

- (i) Johnson James L. *Probability and Statistics for Computer Science*, 6<sup>th</sup> edition, Wiley, 2004.
- (ii) Forsyth David *Probability and Statistics for Computer Science*, 1<sup>st</sup> edition, Springer, 2019.
- (iii) Freund J.E. *Mathematical Statistics with Applications*, 8<sup>th</sup> edition, Pearson Education, 2013.
- (iv) Devore Jay L. *Probability and Statistics for Engineering and the Sciences*, 9<sup>th</sup> 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.

## DSC 07: Data Structures

### Course Objective

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, trees to solve problems. C++ is chosen as the language to understand implementation of these data structures.

### Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Compare two functions for their rates of growth.
2. Understand abstract specification of data-structures and their implementation.
3. Compute time and space complexity of operations on a data-structure.
4. Identify the appropriate data structure(s) for a given application and understand the trade-offs involved in terms of time and space complexity.
5. Apply recursive techniques to solve problems.

### Syllabus

**Unit 1 Growth of Functions, Recurrence Relations:** Functions used in analysis, asymptotic notations, asymptotic analysis, solving recurrences using recursion trees, Master Theorem.

**Unit 2 Arrays, Linked Lists, Stacks, Queues, Deques:** Arrays: array operations, applications, sorting, two-dimensional arrays, dynamic allocation of arrays; Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: stack as an ADT, implementing stacks using arrays, implementing stacks using linked lists, applications of stacks; Queues: queue as an ADT, implementing queues using arrays, implementing queues using linked lists, double-ended queue as an ADT, implementing double-ended queues using linked lists. Time complexity analysis.

**Unit 3 Recursion:** Recursive functions, linear recursion, binary recursion.

**Unit 4 Trees, Binary Trees:** Trees: definition and properties, tree traversal algorithms and their time complexity analysis; binary trees: definition and properties, traversal of binary trees and their time complexity analysis.

**Unit 5 Binary Search Trees, Balanced Search Trees:** Binary Search Trees: insert, delete, search operations, time complexity analysis of these operations; Balanced Search Trees: insert, search operations, time complexity analysis of these operations. Time complexity analysis.

**Unit 6 Binary Heap, Priority Queue:** Binary Heaps: heaps, heap operations, implementing heaps; Priority Queues: priority queue as an ADT. Time complexity analysis.



**Unit 7 Graph Representations and Traversal Algorithms:** Graph as an ADT, data structures for graphs, traversal of graphs, directed graphs. Time complexity analysis.

### References

1. Goodrich, M.T., Tamassia, R., & Mount, D., *Data Structures and Algorithms Analysis in C++*, 2<sup>nd</sup> edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4<sup>th</sup> edition, Prentice Hall of India, 2022.
3. Drozdek, A., *Data Structures and Algorithms in C++*, 4<sup>th</sup> edition, Cengage Learning, 2012.

### Additional references

- (i) Sahni, S. *Data Structures, Algorithms and applications in C++*, 2<sup>nd</sup> edition, Universities Press, 2011.
- (ii) Langsam Y., Augenstein, M. J., & Tanenbaum, A. M. *Data Structures Using C and C++*, Pearson, 2009.

### Suggested Practical List

1. Write a program, using templates, to sort an array of size  $n$ . Give the user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
2. Write a program to implement singly linked list as an ADT that supports the following operations:
  - (i) Insert an element  $x$  at the beginning of the singly linked list
  - (ii) Insert an element  $x$  at  $i^{\text{th}}$  position in the singly linked list
  - (iii) Remove an element from the beginning of the singly linked list
  - (iv) Remove an element from  $i^{\text{th}}$  position in the singly linked list.
  - (v) Search for an element  $x$  in the singly linked list and return its pointer
  - (vi) Concatenate two singly linked lists
3. Write a program to implement doubly linked list as an ADT that supports the following operations:
  - (i) Insert an element  $x$  at the beginning of the doubly linked list
  - (ii) Insert an element  $x$  at  $i^{\text{th}}$  position in the doubly linked list
  - (iii) Insert an element  $x$  at the end of the doubly linked list
  - (iv) Remove an element from the beginning of the doubly linked list
  - (v) Remove an element from  $i^{\text{th}}$  position in the doubly linked list.
  - (vi) Remove an element from the end of the doubly linked list

- (vii) Search for an element x in the doubly linked list and return its pointer
  - (viii) Concatenate two doubly linked lists
4. Write a program to implement circular linked list as an ADT which supports the following operations:
    - (i) Insert an element x at the front of the circularly linked list
    - (ii) Insert an element x after an element y in the circularly linked list
    - (iii) Insert an element x at the back of the circularly linked list
    - (iv) Remove an element from the back of the circularly linked list
    - (v) Remove an element from the front of the circularly linked list
    - (vi) Remove the element x from the circularly linked list
    - (vii) Search for an element x in the circularly linked list and return its pointer
    - (viii) Concatenate two circularly linked lists
  5. Implement a stack as an ADT using Arrays.
  6. Implement a stack as an ADT using the Linked List ADT.
  7. Write a program to evaluate a prefix/postfix expression using stacks.
  8. Implement Queue as an ADT using the circular Arrays.
  9. Implement Queue as an ADT using the Circular Linked List ADT.
  10. Implement Double-ended Queues as an ADT using the Doubly Linked list ADT.
  11. Write a program to implement Binary Search Tree as an ADT which supports the following operations:
    - (i) Insert an element x
    - (ii) Delete an element x
    - (iii) Search for an element x in the BST and change its value to y and then place the node with value y at its appropriate position in the BST
    - (iv) Display the elements of the BST in preorder, inorder, and postorder traversal
    - (v) Display the elements of the BST in level-by-level traversal
    - (vi) Display the height of the BST
  12. Write a program to implement a balanced search tree as an ADT.
  13. Write a Program to implement a priority queue as an ADT using heap data structure.

## **DSC 08: Operating Systems**

### **Course Objective**

The course provides concepts that underlie all operating systems not tied to any particular operating system. The emphasis is done to explain the need and structure of an operating system using its common services such as process management (creation, termination etc.), CPU Scheduling, Process Synchronization, Handling Deadlocks, main memory management, virtual memory, secondary memory management. The course also introduces various scheduling algorithms and structures/techniques used by operating systems to provide these services.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Understand the need of an Operating System & Define Multiprogramming and Multithreading concepts.
2. Implement Process Synchronization service (Critical Section, Semaphores), CPU scheduling service with various algorithms.
3. Learn Main memory Management (Paging, Segmentation) algorithms, Handling of Deadlocks
4. Identify and appreciate the File systems Services, Disk Scheduling service

### **Pre-Requisite**

Familiarity with data structures and any of the high-level languages such as C/C++ is required to do lab assignments.

### **Syllabus**

**Unit 1 Introduction:** Operating Systems (OS) definition and its purpose, Multiprogrammed and Time Sharing Systems, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

**Unit 2 Operating System Structures:** OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

**Unit 3 Process Management :** Process Concept, States, Process Control Block, Process Scheduling, Schedulers, Context Switch, Operation on processes, Threads, Multicore Programming, Multithreading Models, PThreads, Process Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin, Process Synchronization: The critical-section problem and Peterson's Solution, Deadlock characterization, Deadlock handling.

**Unit 4 Memory Management :** Physical and Logical address space, Swapping, Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging.

Virtual Memory Management: Demand Paging and Page Replacement algorithms: FIFO Page Replacement, Optimal Page replacement, LRU page replacement.

**Unit 5 File System:** File Concepts, File Attributes, File Access Methods, Directory Structure: Single-Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories.  
Mass Storage Structure: Magnetic Disks, Solid-State Disks, Magnetic Tapes, Disk Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOK Scheduling.

## References

1. Silberschatz, A., Galvin, P. B., Gagne G. *Operating System Concepts*, 9<sup>th</sup> edition, John Wiley Publications, 2016.

## Additional References

- (i) Dhamdhere, D. M., *Operating Systems: A Concept-based Approach*, 2<sup>nd</sup> edition, Tata McGraw-Hill Education, 2017.
- (ii) Kernighan, B. W., Rob Pike, R. *The Unix Programming Environment*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
- (iii) Stallings, W. *Operating Systems: Internals and Design Principles*, 9<sup>th</sup> edition, Pearson Education, 2018.
- (iv) Tanenbaum, A. S. *Modern Operating Systems*, 3<sup>rd</sup> edition, Pearson Education, 2007.

## Suggested Practical List

1. Execute various LINUX commands for:
  - i. Information Maintenance: wc, clear, cal, who, date, pwd
  - ii. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep, awk
  - iii. Directory Management : cd, mkdir, rmdir, ls
2. Execute various LINUX commands for:
  - i. Process Control: fork, getpid, ps, kill, sleep
  - ii. Communication: Input-output redirection, Pipe
  - iii. Protection Management: chmod, chown, chgrp
3. Write a program(using fork() and/or exec() commands) where parent and child execute:
  - i. same program, same code.
  - ii. same program, different code.
  - iii. before terminating, the parent waits for the child to finish its task.
4. Write a program to report behaviour of Linux kernel including kernel version, CPU type

and model. (CPU information)

5. Write a program to report behaviour of Linux kernel including information on 19 configured memory, amount of free and used memory. (Memory information)
6. Write a program to copy files using system calls.
7. Write a program to implement FCFS scheduling algorithm.
8. Write a program to implement SJF scheduling algorithm.
9. Write a program to implement non-preemptive priority based scheduling algorithm.
10. Write a program to implement SRTF scheduling algorithm.
11. Write a program to calculate sum of n numbers using Pthreads. A list of n numbers is divided into two smaller list of equal size, two separate threads are used to sum the sublists.
12. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

## **DSC 09: Numerical Optimization**

### **Course Objective**

The course aims to provide students an experience of mathematically formulating a large variety of optimization/decision problems emerging out of various fields like data science, machine learning, business and finance. The course focuses on learning techniques to optimize the problems in order to obtain the best possible solution.

### **Course Learning Outcomes**

At the end of the course, students will be able to:

1. Mathematically formulate the optimization problems using the required number of independent variables.
2. Define constraint functions on a problem.
3. Check the feasibility and optimality of a solution.
4. Apply conjugate gradient method to solve the problem.

### **Syllabus**

**Unit 1 Introduction:** Mathematical Formulation using example, Continuous versus Discrete Optimization, Constrained and Unconstrained Optimization, Global and Local Optimization, Stochastic and Deterministic Optimization, Convexity, Optimization Algorithms

**Unit 2 Fundamentals of Unconstrained Optimization:** Concept of a Solution - Recognizing a Local Minimum, Nonsmooth Problems. Overview of Algorithms - Two Strategies: Line Search and Trust Region, Search Directions for Line Search Methods, Models for Trust-Region Methods, Scaling. Line Search - Convergence of Line Search Methods, Rate of Convergence - Convergence Rate of Steepest Descent, Newton's Method, Quasi-Newton Methods. Trust Region - The Cauchy Point algorithm, Global Convergence - Reduction Obtained by the Cauchy Point, Convergence to Stationary Points.

**Unit 3 Conjugate Gradient Methods:** Basic Properties of the Conjugate Gradient Method, A Practical Form of the Conjugate Gradient Method, Rate of Convergence.

**Unit 4 Calculating Derivatives:** Finite-Difference Derivative Approximations, Approximating the Gradient, Approximating a Sparse Jacobian, Approximating the Hessian, Approximating a Sparse Hessian

**Unit 5 Theory of Constrained Optimization:** Local and Global Solutions, Smoothness, Examples - A Single Equality Constraint, A Single Inequality Constraint, Two Inequality Constraints, Tangent Cone and Constraint Qualifications, First-Order Optimality Condition, Second-Order Conditions - Second-Order Conditions and Projected Hessians. Linear and Non-linear Constrained Optimization. Augmented Lagrangian Methods.

## References

1. J. Nocedal and S.J. Wright, Numerical Optimization, Second Edition, Springer Series in Operations Research, 2006.

## DSC 10: Design and Analysis of Algorithms

### Course Objective

The course is designed to develop understanding of different algorithm design techniques and use them for problem solving. The course shall also enable the students to verify correctness of algorithms and analyze their time complexity.

### Course Learning Outcomes

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

1. Compute and compare the asymptotic time complexity of algorithms.
2. Prove correctness of algorithms.
3. Use appropriate algorithm design technique(s) for solving a given problem.
4. Appreciate the difference between tractable and intractable problems.

## Syllabus

**Unit 1 Searching, Sorting, Selection:** Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, Selection Problem, running time analysis and correctness.

**Unit 2 Graphs:** Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering.

**Unit 3 Divide and Conquer:** Introduction to divide and conquer technique, Merge Sort, Quick Sort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

**Unit 4 Greedy algorithms:** Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, etc. with correctness, and analysis of time complexity.

**Unit 5 Dynamic Programming:** Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problem etc., correctness, and analysis of time complexity.

**Unit 6 Intractability:** Concept of polynomial time computation, polynomial time reductions, decision vs optimization problems, Introduction to NP, NP-hard and NP-Complete classes.

**Unit 7 Advanced Analysis Technique:** Amortized Analysis.

## References

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4<sup>th</sup> edition, Prentice Hall of India, 2022.
2. Kleinberg, J., Tardos, E. *Algorithm Design*, 1<sup>st</sup> edition, Pearson, 2013.

## Additional References

- (i) Basse, S., Gelder, A. V., *Computer Algorithms: Introduction to Design and Analysis*, 3<sup>rd</sup> edition, Pearson, 1999.

## Suggested Practical List

1. i. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).  
ii. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).

2. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
3. Write a program to multiply two matrices using the Strassen's algorithm for matrix multiplication.
4. Write a program to sort the elements of an array using Radix Sort.
5. Write a program to sort the elements of an array using Bucket Sort.
6. Display the data stored in a given graph using the Breadth-First Search algorithm.
7. Display the data stored in a given graph using the Depth-First Search algorithm.
8. Write a program to determine a minimum spanning tree of a graph using the Prim's algorithm.
9. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
10. Write a program to solve the weighted interval scheduling problem.
11. Write a program to solve the 0-1 knapsack problem.

For the algorithms at S.No 1 and 2, test run the algorithm on 100 different input sizes varying from 30 to 1000. For each size find the number of comparisons averaged on 10 different input instances; plot a graph for the average number of comparisons against each input size. Compare it with a graph of  $n \log n$ .

## **DSC 11: Database Management System**

### **Course Objective**

The course introduces the students to the fundamentals of database management system and its architecture. Emphasis is given on the popular relational database system including data models and data manipulation. Students will learn about the importance of database structure and its designing using conceptual approach using Entity Relationship Model and formal approach using Normalization. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice of structured query language in a relational database management system and glimpse of basic database administration commands.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:



1. Use database management system software to create and manipulate the database.
2. Create conceptual data models using entity relationship diagrams for modeling real-life situations and designing the database schema.
3. Use the concept of functional dependencies to remove redundancy and update anomalies.
4. Apply normalization theory to get a normalized database scheme.
5. Write queries using relational algebra, a procedural language.
6. Implement relational databases and formulate queries to get solutions of a broad range of data retrieval and data update problems using SQL .
7. Write and execute SQL queries through a high-level language via ODBC connection
8. Database administration commands such as creating/removing users, granting/revoking different privileges to the database users; creating assertions, triggers, and indexes.
9. Learn the importance of index structures and concurrent execution of transactions in database systems.

## **Syllabus**

**Unit 1 Introduction to Database:** Purpose of database system, Characteristics of database approach, data models, database management system, database system architecture, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.

**Unit 2 Entity Relationship Modeling:** Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram notation.

**Unit 3 Relational Data Model:** Update anomalies, Relational Data Model - Concept of relations, schema-instance distinction, keys, relational integrity constraints, referential integrity and foreign keys, relational algebra operators and queries.

**Unit 4 Structured Query Language (SQL):** Querying in SQL, DDL to create database and tables, table constraints, update database-update behaviors, DML, aggregation functions group by and having clauses, retrieve data from the database, generate and query views. Access and manipulate databases using ODBC. Basic Database administration SQL commands.

**Unit 5 Database Design:** Mapping an Entity Relationship model to relational database, functional dependencies and Normal forms, 1NF, 2NF, 3NF and BCNF decompositions and desirable properties of them.

**Unit 6 File indexing and Transaction Processing:** Data Storage and Indexes- Need of file indexes, file organizations, index structures, single- and multi-level indexing, concurrent execution of transactions, ACID properties, need of data recovery and log file.

### References

1. Elmasri, R., Navathe, B. S. *Fundamentals of Database Systems*, 7<sup>th</sup> Edition, Pearson Education, 2015.
2. Krogh, J. W. *MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers*, Apress, 2018.
3. Murach J. *Murach's MySQL*, 3<sup>rd</sup> edition, Pearson, 2019.

### Additional References

- (i) Ramakrishnan, R., Gehrke J. *Database Management Systems*, 3<sup>rd</sup> Edition, McGraw-Hill, 2014.
- (ii) Silberschatz, A., Korth, H. F., Sudarshan S. *Database System Concepts*, 7<sup>th</sup> Edition, McGraw Hill, 2019.
- (iii) Connolly, T. M., Begg, C. E. *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6<sup>th</sup> edition, Pearson, 2019.

### Suggested Practical List

It has three components.

- I. Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor.

<b>STUDENT</b>	<b><u>Roll No</u></b>	<b>StudentName</b>	<b>Course</b>	<b>DOB</b>
	Char(6)	Varchar(20)	Varchar(10)	Date

<b>SOCIETY</b>	<b><u>SocID</u></b>	<b>SocName</b>	<b>MentorName</b>	<b>TotalSeats</b>
	Char(6)	Varchar(20)	Varchar(15)	Unsigned int

<b>ENROLLMENT</b>	<b><u>Roll No</u></b>	<b><u>SID</u></b>	<b>DateOfEnrollment</b>
	Char(6)	Char(6)	<b>Date</b>

Here Rollno (ENROLLMENT) and SID (ENROLLMENT) are foreign keys.

1. Retrieve names of students enrolled in any society.
2. Retrieve all society names.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'
6. Find society details with more than N TotalSeats where N is to be input by the user
7. Update society table for mentor name of a specific society
8. Find society names in which more than five students have enrolled
9. Find the name of youngest student enrolled in society 'NSS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two least popular societies (on the basis of enrolled students)
12. Find the student names who are not enrolled in any society
13. Find the student names enrolled in at least two societies
14. Find society names in which maximum students are enrolled
15. Find names of all students who have enrolled in any society and society names in which at least one student has enrolled
16. Find names of students who are enrolled in any of the three societies 'Debating', 'Dancing' and 'Sashakt'.
17. Find society names such that its mentor has a name with 'Gupta' in it.
18. Find the society names in which the number of enrolled students is only 10% of its capacity.
19. Display the vacant seats for each society.
20. Increment Total Seats of each society by 10%
21. Add enrollment fees paid ('yes'/'No') field in the enrollment table.
22. Update date of enrollment of society id 's1' to '2018-01-15', 's2' to current date and 's3' to '2018-01-02'.
23. Create a view to keep track of society names with the total number of students enrolled in it.
24. Find student names enrolled in all the societies.
25. Count number of societies with more than 5 student enrolled in it
26. Add column Mobile number in student table with default value '9999999999'
27. Find the total number of students whose age is > 20 years.
28. Find names of students who are born in 2001 and are enrolled in at least one society.
29. Count all societies whose name starts with 'S' and ends with 't' and at least 5 students are enrolled in the society.
30. Display the following information:

Society name    Mentor name    Total Capacity    Total Enrolled    Unfilled Seats

II. Do the following database administration commands:  
create user, create role, grant privileges to a role, revoke privileges from a role, create index

III. Execute queries given in part I through a high-level language using ODBC connection.

## **DSC 12: Computer Networks**

### **Course Objective**

The course objectives of this course are to:

understand the concepts behind computer networks and data communication. Learn the use of different layers in standard reference models used for communication. Learn the main features of protocols used at various layers. Understand the utility of different networking devices..

### **Course Learning Outcomes**

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

1. describe the hardware and software components used in a network.
2. compare OSI and TCP/IP reference models at various layers.
3. describe, analyze and compare different data link, network, transport and application layer protocols.
4. design/implement data link and network layer protocols in a simulated networking environment.

### **Syllabus**

**Unit 1 Introduction:** Types of computer networks, Internet, Intranet, network topologies, network classifications. layered architecture approach, OSI Reference Model, TCP/IP Reference Model.

**Unit 2 Physical Layer:** Analog signal, digital signal, digital modulation techniques (ASK, PSK, QAM), encoding techniques, the maximum data rate of a channel, transmission media (guided transmission media, wireless transmission, satellite communication), multiplexing (frequency division multiplexing, time-division multiplexing, wavelength division multiplexing).

**Unit 3 Data Link MAC Layer:** Data link layer services, error detection and correction techniques, error recovery protocols (stop and wait, go back n, selective repeat), multiple access

protocols with collision detection, MAC addressing, Ethernet, data link layer switching, point-to-point protocol.

**Unit 4 Network layer:** Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Routing algorithm (Distance vector and Dijkstra routing), Network Layer protocol- (ARP, IPV4, ICMP, IPV6).

**Unit 5 Transport and Application Layer:** Process to process Delivery- (client-server paradigm, connectionless versus connection-oriented service, reliable versus unreliable); User Datagram Protocols, TCP/IP protocol, Flow Control. FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), Telnet (Remote login protocol), WWW (World Wide Web), HTTP (HyperText Transfer Protocol), URL (Uniform Resource Locator).

## References

1. Tanenbaum, A.S. & Wethrall, D.J, *Computer Networks*, Pearson Education, 2012.
2. Forouzan, B. A., *Data Communication and Networking*, McGraw-Hill Education, 2017.

## Additional References

- (i) Kurose, J. F., & Ross, K. W. *Computer Networking: A Top-Down Approach*, Pearson Education India, 2017.
- (ii) Stallings, W. *Data and Computer Communications*, 10<sup>th</sup> edition, Pearson Education India, 2017.

## Suggested Practical List

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for the noisy channel.
2. Simulate and implement the Stop and Wait protocol for the noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm
6. Simulate and implement the Dijkstra algorithm for shortest-path routing.

## DSC 13: Algorithms and Advanced Data Structures

### Course Objective

This course is designed to build upon the fundamentals in data structures and algorithm design and gain exposure to more data structures and algorithms for new problems.

### Course Learning Outcomes

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

1. Comprehend and use data structures for lists.
2. Use hash tables for dictionaries.
3. Comprehend and use data structures and algorithms for string matching.
4. Comprehend and use disc based data structures.
5. Implement and analyze advanced data structures and algorithms for graphs.
6. Appreciate the strength of randomization in data structures and algorithms.

### Syllabus

**Unit 1 List and Iterator ADTs:** Vectors, Lists, Sequences

**Unit 2 Hash Tables, Dictionaries:** Hash Functions, Collision resolution schemes.

**Unit 3 Strings:** String Matching: KMP algorithm; Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engines

**Unit 4 More on Trees:** 2-4 Trees, B Trees

**Unit 5 More on Graphs:** Bellman Ford Algorithm, Union Find Data Structures - application Kruskal's algorithm

**Unit 6 Randomization:** Randomized Quicksort, Randomized Select, Skip List

**Unit 7 Network Flows:** Ford Fulkerson algorithm for max flow problem.

### References

1. Goodrich, M.T, Tamassia, R., & Mount, D.,. *Data Structures and Algorithms Analysis in C++*, 2<sup>nd</sup> edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4<sup>th</sup> edition, Prentice Hall of India, 2022.
3. Kleinberg, J., Tardos, E. *Algorithm Design*, 1<sup>st</sup> edition, Pearson, 2013.
4. Drozdek, A.,. *Data Structures and Algorithms in C++*, 4<sup>th</sup> edition, Cengage Learning. 2012.

### Suggested Practical List

1. Write a program to sort the elements of an array using Randomized Quick sort (the program should report the number of comparisons).
2. Write a program to find the  $i^{\text{th}}$  smallest element of an array using Randomized Select.

3. Write a program to determine the minimum spanning tree of a graph using Kruskal's algorithm.
4. Write a program to implement the Bellman Ford algorithm to find the shortest paths from a given source node to all other nodes in a graph.
5. Write a program to implement a B-Tree.
6. Write a program to implement the Trie Data structure, which supports the following operations:
  - I. Insert
  - II. Search
7. Write a program to search a pattern in a given text using the KMP algorithm.
8. Write a program to implement a Suffix tree.

## **DSC 14: Theory of Computation**

### **Course Objective**

This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine; and their relationships with formal languages. Students will also learn about the limitations of computing machines as this course addresses the issue of which problems can be solved by computational means (decidability vs undecidability). The course is aimed to -

1. formalize the notion of problems via formal languages
2. make students aware of the notion of computation using "abstract computing devices" called automata
3. familiarize students with the hierarchy of classes of problems or formal languages (regular, context-free, context-sensitive, decidable, and undecidable) and the hierarchy of classes of automata (finite automata, pushdown automata, and Turing machines)

### **Course Learning Outcomes**

On successful completion of the course, a student will be able to:

1. design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.
2. apply pumping lemma to prove that a language is non-regular/non-context-free.
3. describe limitations of a computing machines and
4. recognize what can be solved and what cannot be solved using these machines.

## Pre-requisites

A course in programming language, Design and Analysis of Algorithms

## Syllabus

**Unit I Introduction:** Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

**Unit II Finite Automata and Regular:** Regular expressions, Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NFA), relationship between NFA and DFA, Transition Graphs (TG), properties of regular languages, the relationship between regular languages and finite automata, pumping lemma, Kleene's theorem.

**Unit III Context-Free Languages (CFL):** Context-Free Grammars (CFG), deterministic and non-deterministic Pushdown Automata (PDA), relationship between CFG and PDA, parse trees, leftmost derivation, Ambiguities in grammars, pumping lemma for CFL, properties of CFL, Chomsky Normal Form.

**Unit IV Turing Machines and Models of Computations:** Turing machine as a model of computation, configuration of Turing machine, Recursive and recursively enumerable languages, Church Turing Thesis, Universal Turing Machine, decidability, Halting problem.

## References

1. Michael Sipser, *Introduction to the Theory of Computation*, Cengage, 2014
2. Harry R. Lewis and Christos H. Papadimitriou, *Elements of the Theory of Computation*, 2<sup>nd</sup> Edition, Prentice Hall of India (PHI), 2002
3. Daniel I.A. Cohen, *Introduction to Computer Theory*, 2<sup>nd</sup> Edition, Wiley India Pvt. Ltd., 2011.

## Additional References

- (i) J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 3<sup>rd</sup> edition, Addison Wesley, 2006.
- (ii) Peter Linz, *An Introduction to Formal Languages and Automata*, 6<sup>th</sup> edition, Jones & Bartlett Learning, 2017.



## Course Objective

This course will provide fundamental approaches and techniques used to develop good quality software. This includes learning of various software development process frameworks, requirement analysis, design modeling, qualitative and quantitative software metrics, risk management, and testing techniques.

## Course Learning Outcomes

On successful completion of the course, a student will be able to:

1. Understand the software development models.
2. Analyze and model customer requirements and build design models.
3. Estimate and prepare schedule for software projects.
4. Analyze the impact of risks involved in software development.
5. Design and build test cases, and to perform software testing.

## Pre-requisites:

A course in programming language

## Syllabus

**Unit I Introduction:** Software Engineering - A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile, Agile Model – Scrum. [9 Hours]

**Unit II Software Requirements Analysis and Specification:** Use Case Approach, Software Requirement Specification Document, Flow-oriented Model, Data Flow Model. [6 Hours]

**Unit III Design Modeling:** Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Structure Charts. [8 Hours]

**Unit IV Software Metrics and Project Estimation:** Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations); Project Scheduling (Timeline charts, tracking the schedule). [7 Hours]

**Unit V Quality Control and Risk Management:** Quality Control and Quality Assurance, Software Process Assessment and Improvement; Software Risks, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management. [5 Hours]

**Unit VI Software Testing:** Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing. [10 Hours]

### References

1. Pressman, R.S. *Software Engineering: A Practitioner's Approach*, 9<sup>th</sup> edition, McGraw-Hill, 2020.
2. Aggarwal, K.K., Singh, Y. *Software Engineering*, 3<sup>rd</sup> edition, New Age International Publishers, 2007.

### Additional References

- (i) Jalote, P. *An Integrated Approach to Software Engineering*, 3<sup>rd</sup> Edition, Narosa Publishing House, 2005.
- (ii) Sommerville, I. *Software Engineering*, 9<sup>th</sup> edition, Addison Wesley, 2011.
- (iii) The Definitive Guide to Scrum: The Rules of the Game, Ken Schwaber, Jeff Sutherland, July 2016.

### Suggested Practical list

The students document, design and code a module of a Software Project using an appropriate Software Process model. The Software Project should address the following concepts of Software Engineering.

1. Problem Statement, Process Model
2. Requirement Analysis: Create Data Flow, Data Dictionary, Use Cases, Sequence Diagram, Software Requirement Specification Document
3. Project Management: Timeline Chart, Compute FP, Effort, Cost, Risk Table.
4. Design Engineering: Architectural Design, Pseudocode of a small module.
5. Coding: Develop at least a single module using any programming Language

6. Testing: Compute Basis path set for at least a single module from the project, Generate test cases.

Some of the Sample Projects are given below though they are not limited to this.

1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers
2. DTC Route Information: Online information about the bus routes and their frequency and fares.
3. Car Pooling: To maintain a web-based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation System
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

## **DSC 16 : Artificial Intelligence**

### **Course Objective**

The course objectives of this course are to:

Understand the foundations, basic concepts and techniques of Artificial Intelligence (AI).

Apply informed search techniques for different applications.

Impart knowledge about the use of core AI techniques having applicability to a wide range of real-world problems.

Learn about various knowledge representation techniques and writing Prolog programs.

Learn about the latest techniques for developing AI systems.

### **Course Learning Outcomes**

On successful completion of this course, students will be able to:

1. identify problems that are amenable to solutions by specific AI methods.
2. appreciate the utility of different types of AI agents.
3. apply different informed search techniques for solving real world problems.
4. use knowledge representation techniques for AI systems..
5. understand human level, data driven and end to end approaches to AI.

### **Prerequisite:**

Knowledge of any Programming Language, Basics of algorithms, Data Structures, and Theory of Computation.

### **Syllabus**

**Unit 1 Introduction:** Introduction to artificial intelligence, background and applications, Turing test, Weak AI, Strong AI, Narrow AI, Artificial General Intelligence, Super AI, rational agent approaches to AI, introduction to intelligent agents, their structure, behavior and task environment.

**Unit 2 Problem Solving and Searching Techniques:** Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A\* algorithm, constraint satisfaction problem, means-end analysis, introduction to game playing, min-max and alpha-beta pruning algorithms.

**Unit 3 Knowledge Representation:** Propositional logic, First-Order Predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, Introduction to Programming in Logic (PROLOG).

**Unit 4 Understanding Natural Languages:** Components and steps of communication, the contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented transition nets.

**Unit 5 AI The Present and the Future:** Symbolic AI, Data-driven AI and Machine Learning, Bias-Variance tradeoff in Machine Learning, End-to-end Deep Learning based AI, some applications of symbolic and data driven AI, Interpretable and Explainable AI, Ethics of AI: benefits and risks of AI.

### **Text Books**

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4<sup>th</sup> edition, 2020.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3<sup>rd</sup> edition, Tata McGraw Hill, 2010.
3. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4<sup>th</sup> edition, 2012.

### **Reference Books**

- (i) Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
- (ii) Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
- (iii) Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5<sup>th</sup> edition, Springer, 2003.

### **Suggested Practical List**

1. Write a program to implement two agents to communicate with each other with message passing in Python/Prolog.
2. Write a program to implement the Hill climbing search algorithm in Python/Prolog.
3. Write a program to implement the Best first search algorithm in Python/Prolog.
4. Write a program to implement A\* search algorithm in Python/Prolog.
5. Write a program to implement the min-max search algorithm in Python/Prolog.
6. Write a program to solve the Water-Jug Problem in Python/Prolog.
7. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Python/Prolog.
8. Write a prolog program to implement the family tree and demonstrate the family relationship.
9. Write a prolog program to implement knowledge representation using frames with appropriate examples.

10. Write a Prolog program to implement `conc(L1, L2, L3)` where L2 is the list to be appended with L1 to get the resulted list L3.
11. Write a Prolog program to implement `reverse(L, R)` where List L is original and List R is reversed list.
12. Write a PROLOG program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
13. Write a prolog program to recognize context free grammar  $a^n b^n$ .
14. Using any sample dataset, implement any classification problem in Python language.
15. In the above classification problem (program 14), generate explanations for the decision taken using LIME tool.

## **DSC 17: Machine Learning**

### **Course Prerequisites**

Mathematics for Computing (I II and III)

### **Course Objective**

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

### **Course Learning Outcomes**

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

1. Differentiate between supervised and unsupervised learning tasks.
2. Appreciate the need of preprocessing, feature scaling and feature selection.
3. Understand the fundamentals of classification, regression and clustering
4. Implement various machine learning algorithms learnt in the course.

### **Syllabus**

**Unit 1 Introduction:** Basic definitions and concepts, key elements, supervised and unsupervised learning, introduction to reinforcement learning, applications of ML.

**Unit 2 Preprocessing:** Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis).

**Unit 3 Regression:** Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

**Unit 4 Classification:** Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics.

**Unit 5 Clustering:** Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

## References

1. Mitchell, T.M. *Machine Learning*, McGraw Hill Education, 2017.
2. James, G., Witten. D., Hastie. T., Tibshirani., R. *An Introduction to Statistical Learning with Applications in R*, Springer, 2014.
3. Alpaydin, E. *Introduction to Machine Learning*, MIT press, 2009.

## Additional References

- (i) Flach, P., *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, 2015.
- (ii) Christopher & Bishop, M., *Pattern Recognition and Machine Learning*, New York: Springer-Verlag, 2016.
- (ii) Sebastian Raschka, *Python Machine Learning*, Packt Publishing Ltd, 2019.

## Suggested Practical List

Use Python for practical labs for Machine Learning. Utilize publically available datasets from online repositories like <https://data.gov.in/> and <https://archive.ics.uci.edu/ml/datasets.php>

For evaluation of the regression/classification models, perform experiments as follows:

- Scale/Normalize the data

- Reduce dimension of the data with different feature selection techniques
- Split datasets into training and test sets and evaluate the decision models
- Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows:

- MSE and  $R^2$  score for regression models
- Accuracy, TP, TN, FP, FN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier
2. Simple Linear Regression multiple linear regression
3. Polynomial Regression
4. Lasso and Ridge Regression
5. Logistic regression
6. Artificial Neural Network
7.  $k$ -NN classifier
8. Decision tree classification
9. SVM classification
10. K-Means Clustering
11. Hierarchical Clustering

## **DSC 18: Introduction to Parallel Programming**

### **Course Prerequisites**

Programming in C++, Data Structures, Computer System Architecture, Operating Systems

### **Course Objective**

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

### **Course Learning Outcomes**

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

1. Appreciate the need of Parallel algorithms
2. Describe architectures for parallel and distributed systems.
3. Develop elementary parallel algorithms in shared memory models.
4. Develop elementary parallel algorithms in distributed memory models.



## Syllabus

**Unit I Introduction to Parallel Computing:** Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

**Unit II OpenMP programming for shared memory systems:** Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model, Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks, Recursive task spawning and pitfalls

**Unit III MPI programming for distributed memory systems:** MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, MPI Collectives (MPI Collectives and MPI broadcast, MPI Gathering and scattering collectives, MPI reduction and Alltoall collectives, MPI collectives design), Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

**Unit IV Applications:** Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm

## References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2<sup>nd</sup> edition, Addison-Wesley, 2003.
2. Quinn, M. *Parallel Programming in C with MPI and OpenMP*, 1<sup>st</sup> Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016.

## Suggested Practical List

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP

2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP
4. Implement Dijkstra's Algorithm in parallel using OpenMP

## **DSC 19: Compiler Design**

### **Course Objective**

The basic objective of the compiler design course is to understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler. It also aims to understand the use of basic compiler building tools.

### **Course Learning Outcomes**

On successful completion of the course, the students will be able to:

1. Explain the concepts and different phases of compilation.
2. Represent language tokens using regular expressions and context free grammars.
3. Describe the working of lexical analyzers.
4. Understand the working of different types of parsers and parse a particular string.
5. Describe intermediate code representations using syntax trees and DAG's as well as use this knowledge to generate intermediate code in the form of three address code representations.
6. Apply optimization techniques to intermediate code and generate machine code for high level language program.
7. Use Lex and Yacc automated compiler generation tools.

### **Syllabus**

**Unit 1 Introduction:** Overview of compilation, Phases of a compiler.

**Unit 2 Lexical Analysis:** Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, Error reporting, Regular expressions and definitions , Lexical Analyzer Generator-Lex.

**Unit 3 Syntax Analysis:** CFGs, left recursion, left factoring, Top-down parsing- LL parser, Bottom-up parsing- LR parser, Parser Generator-yacc.

**Unit 4 Intermediate representations:** Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, loops and conditional statements, Type Checking.

**Unit 5 Storage organization & Code generation:** Activation records, stack allocation, Issues in Code Generation – Design of a simple Code Generator.

**Unit 6 Code optimization :** Principal sources of optimization, Peephole optimization.

## References

1. Aho, A., Lam, M., Sethi, R., & Ullman, J. D. *Compilers: Principles, Techniques, and Tools*, 2<sup>nd</sup> edition, Addison Wesley, 2006.

## Additional References

- (i) V Raghvan, *Principles of Compiler Design*, TMH, 2010.
- (ii) Santanu Chattopadhyay, *Compiler Design*, PHI, 2005.

## Suggested Practical List

1. Write a Lex program to count the number of lines and characters in the input file.
2. Write a Lex program to count the number of vowels and consonants in a given string
3. Write a Lex program that implements the Caesar cipher: it replaces every letter with the one three letters after in alphabetical order, wrapping around at Z. e.g. a is replaced by d, b by e, and so on z by c.
4. Write a Lex program that finds the longest word (defined as a contiguous string of upper and lower case letters) in the input.
5. Write a Lex program that distinguishes keywords, integers, floats, identifiers, operators, and comments in any simple programming language.
6. Write a Lex program to count the number of words, characters, blank spaces and lines in a C file.
7. Write a Lex specification program that generates a C program which takes a string “abcd” and prints the following output  
abcd  
abc  
a
8. Write a Lex program to recognize a valid arithmetic expression.
9. Write a YACC program to find the validity of a given expression (for operators + - \* and /)A program in YACC which recognizes a valid variable which starts with a letter followed by a digit. The letter should be in lowercase only.

10. Write a program in YACC to evaluate an expression (simple calculator program for addition and subtraction, multiplication, division).
11. Write a program in YACC to recognize the string „abbb“, „ab“, „a“ of the language (an b n , n>=1).
12. Write a program in YACC to recognize the language (an b , n>=10). (output to say input is valid or not)

### **Additional Suggestive list of Practicals (Can be implemented in C++/Python)**

1. Write a program to implement DFAs that recognize identifiers, constants, and operators of the mini language.
2. Write a program Design a Lexical analyzer for the above language. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments. Identifiers may be of restricted length.
3. Write a program to check the types of expressions in a language.
4. Write a translator to translate a 3-address code into assembly code.

## **DSC 20: Information Security**

### **Course Objective**

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures. This course also touches upon the implications of security in cloud and Internet of Things (IoT).

### **Course Learning Outcomes**

On successful completion of this course, a student will be able to

1. Identify the major types of threats to information security.
2. Describe the role of cryptography in security.
3. Discover the strengths and weaknesses of private and public key cryptosystems.
4. Identify and apply various access control and authentication mechanisms.
5. Discuss data and software security and related issues.
6. Explain network security threats and attacks.
7. Articulate the need for security in cloud and IoT.

## Syllabus

**Unit 1 Overview:** Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles, Attack Surfaces and Attack Trees.

**Unit 2 Cryptographic Techniques:** Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

**Unit 3 Data Security:** User authentication and Access Control, Database and Data Center Security

**Unit 4 Software Security:** Types of Malicious Software, Threats, Viruses, Worms, SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing, Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. Overflow Attacks - Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks. Handling Program Input, Writing Safe Program Code, Interacting with the Operating System and Other Programs.

**Unit 5 Network Security:** Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, Firewalls, Secure Email and S/MIME, Secure Sockets Layer (SSL) and Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security, Public-Key Infrastructure.

**Unit 6 Wireless, Cloud and IoT Security:** Cloud Computing, Cloud Security Concepts, Cloud Security Approaches, The Internet of Things, IoT Security. Wireless Security Overview, Mobile Device Security, IEEE 802.11 Wireless LAN.

## References

1. Stallings, W. and Brown L., *Computer Security: Principles and Practice*, 4<sup>th</sup> edition, Pearson Education, 2018.

## Additional References

- (i) Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5<sup>th</sup> edition, Prentice Hall, 2015.
- (ii) Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications*, 2<sup>nd</sup> edition, Pearson Education, 2004.
- (iii) Stallings W. *Cryptography and network security*, 7<sup>th</sup> edition, Pearson Education, 2018.

- (iv) Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.
- (v) Stallings W. *Network security essentials Applications and Standards*, 6<sup>th</sup> edition, Pearson Education, 2018.
- (vi) Whitman M.E., Mattord H.J., *Principle of Information Security*, 6<sup>th</sup> edition, Cengage Learning, 2017.
- (vii) Bishop M., *Computer Security: Art and Science*, 2<sup>nd</sup> Revised Edition, Pearson Education, 2019.
- (viii) Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2<sup>nd</sup> Edition, John Wiley & Sons, 2008.

### **Suggested Practical List**

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.
12. Illustrate the Ciphertext only and Known plaintext attacks.
13. Implement a stream cipher technique.

**All DSEs are available as Generic Electives to students of non-Computer Science stream.**

**List of Pathways in Discipline Specific Electives (DSEs)**

1. [Data Science](#)
2. [Image Processing](#)
3. [Theoretical Computer Science](#)
4. [AI and Machine Learning](#)
5. [Cyber Security](#)
6. [Networks](#)
7. [Other DSEs](#)

Sem	Data Science	Image Processing	Theoretical Computer Science	AI and Machine Learning	Cyber Security	Networks	Other DSEs
III Choose one from the pool of courses	<a href="#">Data analysis and Visualization</a> (DAV)				Cryptography		<a href="#">Microprocessor</a>
IV Choose one from the pool of courses	<a href="#">Data Mining-I</a>		<a href="#">Combinatorial Optimisation</a>		Data Privacy		<a href="#">Introduction to Web Programming</a>
V Choose one from the pool of courses	<a href="#">Data Mining-I</a>				Network Security	Unix Network Programming	
VI	<a href="#">Social</a>	<a href="#">Compute</a>		<a href="#">Deep</a>	Ethical	Wireless	Research

Choose one from the pool of courses	<a href="#">Network Analytics</a>	<a href="#">r Graphics</a>		<a href="#">Learning</a>	Hacking	Technology & Standards	Methodology (RM)
VII Choose 1-3 courses from the pool of courses		<a href="#">Digital Image Processing</a>	<a href="#">Advanced Algorithms</a>	<a href="#">Reinforcement Learning</a>	Cyber Forensics	Advanced TCP/IP  Mobile and Satellite Communication	Research Methodology (RM)
VIII Choose 1-3 courses from the pool of courses		<a href="#">Information and Image Retrieval</a>		<a href="#">Natural Language Processing</a>	IT Infrastructure Management / Critical Infrastructure Management <a href="#">Blockchain and Its applications</a>	Cloud Computing  IoT	<a href="#">Distributed Algorithms</a>

**Note:**

1. Research Methodology course is to be done exactly once, either in Semester VI or in Semester VII.
2. For a prerequisite of C++/Python/Java, a course at plus 2 level is acceptable.
3. Batch size for Practicals will be (8-10) and Tutorials will be (12-15).



**Table of DSE Courses (Data Science)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
III		<a href="#">Data analysis and Visualization (DAV)</a>	3	0	1	4	Programming using Python, Probability and Statistics, Linear Algebra
IV		<a href="#">Data Mining-I</a>	3	0	1	4	Programming using Python, Probability and Statistics, Linear Algebra, Data analysis and visualization
V		<a href="#">Data Mining-II</a>	3	0	1	4	Programming using Python, Probability and Statistics, Linear Algebra, Data analysis and visualization, Data Mining I
VI		<a href="#">Social Network Analytics</a>	3	1	0	4	Programming

							ng using Python, Probability and Statistics, Linear Algebra
--	--	--	--	--	--	--	---

**Table of DSE Courses (Image Processing)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
VI		<a href="#">Computer Graphics</a>	3	0	1	4	Mathematics for Computing - I, Object Oriented Programming with C++
VII		<a href="#">Digital Image Processing</a>	3	0	1	4	Programming using Python, Mathematics for Computing - I
VIII		<a href="#">Information and Image Retrieval</a>	3	0	1	4	Digital Image Processing

**Table of DSE Courses (Theoretical Computer Science)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
IV		<a href="#">Combinatorial Optimisation</a>	3	1	0	4	None
VII		<a href="#">Advanced Algorithms</a>	3	1	0	4	Discrete Structures, Design and Analysis of Algorithms, Combinatorial Optimization

**Table of DSE Courses (AI and Machine Learning)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
VI		<a href="#">Deep Learning</a>	3	0	1	4	Probability and Statistics, Linear Algebra, Machine Learning
VII		<a href="#">Reinforcement Learning</a>	3	1	0	4	Machine Learning
VIII		<a href="#">Natural Language Processing</a>	3	0	1	4	Machine Learning

**Table of DSE Courses (Cyber Security)  
(Syllabus under Preparation)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
III		Cryptography					Discrete Mathematics
IV		Data Privacy	3	0	1	4	None
V		Network Security	3	0	1	4	Computer Networks, C/C++
VI		Ethical Hacking	3	0	1	4	Operating System, C/C++, Network Security
VII		Cyber Forensics	3	0	1	4	Operating Systems, Network Security, C/C++
VIII		IT Infrastructure Management / Critical Infrastructure Management	3	0	1	4	None
VIII		<a href="#">Blockchain and Its applications</a>	3	0	1	4	Data Structures, C++, Cryptography

--	--	--	--	--	--	--	--

**Table of DSE Courses (Networks)  
(Syllabus under Preparation)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
V		Unix Network Programming	3	0	1	4	Knowledge of any programming language. Computer Networks
VI		Wireless Technology & Standards	3	0	1	4	Computer Networks
VII		Advanced TCP/IP	3	0	1	4	Computer Networks
VII		Mobile and Satellite Communication	3	0	1	4	Computer Networks
VIII		Cloud Computing	3	0	1	4	Computer Networks
VIII		IoT	3	0	1	4	Computer Networks

**Table of DSE Courses (Others)  
(Syllabus Under Preparation)**

Semester	DSE -No.	Title	L	T*	P*	Total credits	Pre-requisites
III		<a href="#">Microprocess</a> <a href="#">or</a>	3	0	1	4	None
IV		<a href="#">Introduction to Web Programming</a>	3	0	1	4	None
VI		Research Methodology					
VII		Research Methodology					
VIII		<a href="#">Distributed Algorithms</a>	3	0	1	4	C++, Data Structures and Algorithms-1, Operating Systems, Design and Analysis of Algorithms, Theory of Computation

**Pathway: Data Science**

**DSE: Data Analysis and Visualization (DAV)**

**Course Prerequisites:** Programming using Python, Probability and Statistics, Linear Algebra

## **Course Objective**

This course is designed to introduce the students to real-world data analysis problems, use of statistics to get a deterministic view of data and interpret results in the field of exploratory data science using Python. This course is the first in the “Data Science” pathway and builds the foundation of three subsequent courses in the pathway.

## **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Apply descriptive statistics to obtain a deterministic view of data
2. Perform data handling using Numpy arrays
3. Load, clean, transform, merge and reshape data using Pandas
4. Visualize data using Pandas and matplotlib libraries
5. Solve real world data analysis problems

## **Syllabus**

**Unit 1 Introduction to basic statistics and analysis:** Fundamentals of Data Analysis, Statistical foundations for Data Analysis, Types of data, Descriptive Statistics, Correlation and covariance, Linear Regression, Statistical Hypothesis Generation and Testing, Python Libraries: NumPy, Pandas, Matplotlib

**Unit 2 Array manipulation using Numpy:** Numpy array: Creating Numpy arrays; various data types of Numpy arrays, indexing and slicing, swapping axes, transposing arrays, data processing using Numpy arrays

**Unit 3 Data Manipulation using Pandas:** Data Structures in Pandas: Series, DataFrame, Index objects, Loading data into Pandas data frame, Working with DataFrames: Arithmetics, Statistics, Binning, Indexing, Reindexing, Filtering, Handling missing data, Hierarchical indexing, Data wrangling: Data cleaning, transforming, merging and reshaping

**Unit 4 Plotting and Visualization:** Using Matplotlib to plot data: figures, subplots, markings, color and line styles, labels and legends, Plotting functions in Pandas: Line, bar, Scatter plots, histograms, stacked bars, Heatmap

**Unit 5 Data Aggregation and Group operations:** Group by Mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation

## References

1. McKinney W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython*, 2<sup>nd</sup> edition, O'Reilly Media, 2018.
2. Molin S. *Hands-On Data Analysis with Pandas*, Packt Publishing, 2019.
3. Gupta S.C., Kapoor V.K. *Fundamentals of Mathematical Statistics*, 12<sup>th</sup> edition, Sultan Chand & Sons, 2020.

## Additional References

- (i) Chen D. Y. *Pandas for Everyone: Python Data Analysis*, 1<sup>st</sup> edition, Pearson Education, 2018.
- (ii) Miller J.D. *Statistics for Data Science*, Packt Publishing Limited, 2017.

## Suggested Practical List

Use a dataset of your choice from Open Data Portal ([https:// data.gov.in/](https://data.gov.in/), UCI repository) or load from scikit, seaborn library for the following exercises to practice the concepts learnt.

1. Load a Pandas dataframe with a selected dataset. Identify and count the missing values in a dataframe. Clean the data after removing noise as follows
  - a) Drop duplicate rows.
  - b) Detect the outliers and remove the rows having outliers
  - c) Identify the most correlated positively correlated attributes and negatively correlated attributes
2. Import iris data using sklearn library or (Download IRIS data from: <https://archive.ics.uci.edu/ml/datasets/iris> or import it from sklearn.datasets)
  - i. Compute mean, mode, median, standard deviation, confidence interval and standard error for each feature
  - ii. Compute correlation coefficients between each pair of features and plot heatmap
  - iii. Find covariance between length of sepal and petal
  - iv. Build contingency table for class feature
3. Load Titanic data from sklearn library , plot the following with proper legend and axis labels:
  - a. Plot bar chart to show the frequency of survivors and non-survivors for male and female passengers separately
  - b. Draw a scatter plot for any two selected features



- c. Compare density distribution for features age and passenger fare
  - d. Use a pair plot to show pairwise bivariate distribution
4. Using Titanic dataset, do the following
    - a. Find total number of passengers with age less than 30
    - b. Find total fare paid by passengers of first class
    - c. Compare number of survivors of each passenger class
  5. Download any dataset and do the following
    - a. Count number of categorical and numeric features
    - b. Remove one correlated attribute (if any)
    - c. Display five-number summary of each attribute and show it visually

**Project:** Students are encouraged to work on a good dataset in consultation with their faculty and apply the concepts learned in the course.

### **Pathway: Data Science**

### **DSE: Data Mining-I**

**Course Prerequisites:** Programming using Python, Probability and Statistics, Linear Algebra, Data analysis and visualization

### **Course Objective**

This course aims to introduce data mining techniques and their application on real-life datasets. The students will learn to pre-process the dataset and make it ready for application of data mining techniques. The course will focus on three main techniques of data mining i.e. Classification, Clustering and Association Rule Mining. Different algorithms for these techniques will be discussed along with appropriate evaluation metrics to judge the performance of the results delivered.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to :

1. Pre-process the data for subsequent data mining tasks

2. Apply a suitable classification algorithm to train the classifier and evaluate its performance.
3. Apply appropriate clustering algorithm to cluster the data and evaluate clustering quality
4. Use association rule mining algorithms and generate frequent item-sets and association rules

## Syllabus

**Unit 1 Introduction to Data Mining:** Motivation and challenges for data mining, Types of data mining tasks, Applications of data mining, Data measurements, Data quality, Supervised vs. unsupervised techniques

**Unit 2 Data Pre-processing:** Data aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, variable transformation.

**Unit 3 Cluster Analysis:** Basic concepts of clustering, measure of similarity, types of clusters and clustering methods, K-means algorithm, measures for cluster validation, determine optimal number of clusters

**Unit 4 Association rule mining:** Transaction data-set, frequent itemset, support measure, rule generation, confidence of association rule, Apriori algorithm, Apriori principle

**Unit 5 Classification:** Naive Bayes classifier, Nearest Neighbour classifier, decision tree, overfitting, confusion matrix, evaluation metrics and model evaluation.

## References

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. *Introduction to Data Mining*, 2<sup>nd</sup> edition, Pearson, 2021.
2. Han J., Kamber M. and Pei J. *Data Mining: Concepts and Techniques*, 3<sup>rd</sup> edition, 2011, Morgan Kaufmann Publishers.
3. Zaki M. J. and Meira J. Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2<sup>nd</sup> edition, Cambridge University Press, 2020.

## Additional References

- (i) Aggarwal C. C. *Data Mining: The Textbook*, Springer, 2015.
- (ii) Dunham M. *Data Mining: Introductory and Advanced Topics*, 1<sup>st</sup> edition, Pearson Education India, 2006.

## Suggested Practical List

All topics covered in theory will be implemented using Python. The operations may be performed on the datasets loaded through scikit, seaborn libraries or can be downloaded from Open Data Portal ([https:// data.gov.in/](https://data.gov.in/), UCI repository <http://archive.ics.uci.edu/ml/> ).

### **Recommended Datasets for :**

**Classification:** Abalone, Artificial Characters, Breast Cancer Wisconsin (Diagnostic)

**Clustering:** Grammatical Facial Expressions, HTRU2, Perfume data

**Association Rule Mining:** MovieLens, Titanics

### **Additional Suggested Practicals List**

1. Apply data cleaning techniques on any dataset (e.g, wine dataset). Techniques may include handling missing values, outliers, inconsistent values. A set of validation rules can be prepared based on the dataset and validations can be performed.
2. Apply data pre-processing techniques such as standardization/normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset
3. Run Apriori algorithm to find frequent itemsets and association rules on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns
  - a) Use minimum support as 50% and minimum confidence as 75%
  - b) Use minimum support as 60% and minimum confidence as 60 %
4. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations:
  - I. a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3%
  - II. Training set is chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained.  
Data is scaled to standard format.
5. Use Simple K-means algorithm for clustering on any dataset. Compare the performance of clusters by changing the parameters involved in the algorithm. Plot MSE computed after each iteration using a line plot for any set of parameters.

**Project:** Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

## **Pathway: Data Science**

### **DSE: Data Mining-II**

**Course Prerequisites:** Programming using Python, Probability and Statistics, Linear Algebra, Data analysis and visualization, Data Mining-I

#### **Course Objective**

The course introduces the students to the important supervised and unsupervised learning techniques. Students will learn about the importance of ensemble methods, cluster analysis, anomaly detection and their applicability in mining patterns in real applications. At the end students will be exposed to two advanced topics: text mining and time-series mining. Students will use the learned topics in solving real applications using Open-source softwares.

#### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Differentiate between partition-based, density-based and hierarchical clustering
2. Build ensemble models to improve predictive performance of the classifier
3. Identify anomalies and outliers using supervised and unsupervised techniques
4. Analyze time-series data and extract patterns from the stamped data
5. Mine textual data and do topic modeling

#### **Syllabus**

**Unit 1 Clustering:** Partitioning Methods, Hierarchical Methods, Density-Based Methods, Comparison of different methods

**Unit 2 Ensemble Methods:** Need of ensemble, Random Forests, Bagging and Boosting

**Unit 3 Anomaly Detection:** Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-based and density-based outlier detection, Clustering-based approaches

**Unit 4 Mining Text Data:** Document Preparation and Similarity, Clustering Methods for Text, Topic Modeling

**Unit 5 Stream Mining:** Time series basics, Date Ranges, Frequencies, and Shifting, Resampling and moving windows functions, Decay function, Clustering stamped data: STREAM and CluStream

## References

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. *Introduction to Data Mining*, 2<sup>nd</sup> edition, Pearson, 2019.
2. Zaki M. J. and Meira J. Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2<sup>nd</sup> edition, Cambridge University Press, 2020.
3. Aggarwal C. C. *Data Mining: The Textbook*, Springer, 2015.

## Additional References

- (i) Han J. Kamber M. and Pei J. *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2011.
- (ii) Dunham M. *Data Mining: Introductory and Advanced Topics*, Pearson, 2006.

## Suggested Practical List

1. Apply Partitioning Methods, Hierarchical Methods, Density-Based Methods for clustering on a data set and compare the performance of the obtained results using different metrics
2. Create an ensemble using Random Forest and show the impact of bagging and boosting on the performance
3. Apply different outlier-detection methods on a noisy dataset and compare their effectiveness in terms of outliers reported
4. Compute similarity between two documents after required document preparation
5. Considering a time-stamped data (sales data/weather data), compare the aggregate values visually using different moving windows function
6. Write a program to find the latent topics in a document using any topic modeling method and display top 5 terms that contribute to each topic along with their strength. Also, visualize the distribution of terms contributing to the topics.

**Project:** Students should be promoted to take up one project covering at least one unit of the syllabus on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

## Pathway: Data Science

### DSE: Social Network Analytic

**Course Prerequisites:** Programming using Python, Probability and Statistics, Linear Algebra

## Course Objective

The course introduces basic graph theory and draws distinction between graph as an abstract structure and real-life situation modeled as network. This course aims to expose the students to the strengths and capabilities of network analysis and their applications through the use of open source softwares.

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Model real life situation as networks
2. Identify and apply quantitative network measures to characterize social networks at the local and global level
3. Generate synthetic networks that satisfy properties of real world networks
4. Discover, analyze and evaluate the intrinsic community structure of networks
5. Model an information diffusion process for predictive analysis of networks

## Syllabus

**Unit 1 Introduction to Social Network Analysis:** Graph theory, random walk, degree distribution, mapping of real world situation into networks and applications of social network analysis, types of networks

**Unit 2 Network Measures:** Centrality measures, Page Rank, Hubs and Authority, Assortativity, Transitivity and Reciprocity, Similarity and Structural Equivalence

**Unit 3 Network Models:** Properties of Real-World Networks, Random Network Model, Small World Network Model, Preferential Attachment Model

**Unit 4 Community Structure in Networks:** Types of Communities, Community Detection algorithms and evaluation of communities obtained

**Unit 5 Information Diffusion in Social Media:** Information Cascades, Diffusion of Innovations, Basic Epidemic Models

## References

1. Chakraborty T. *Social Network Analysis*, 1<sup>st</sup> edition, Wiley India Pvt. Ltd., 2021.
2. Zafarani R., Abbasi M. A., Liu H. *Social Media Mining: An Introduction*, 1<sup>st</sup> edition, Cambridge University Press, 2014.

3. Barabási A. L. , Pósfai M. *Network Science*, 1<sup>st</sup> edition, Cambridge University Press, 2016.

### **Additional References**

- (i) Easley, Kleinberg J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, 1<sup>st</sup> edition, Cambridge University Press, 2012.

### **Suggested Practical List**

#### **Python Packages like igraph, NetworkX, NDlib etc. may be used for programming**

1. Plot a weighted directed network such that node size and edge width is proportional to their degree and edge weight respectively
2. Compute and plot degree distribution of a real-world network. Also compute its local and global properties.
3. Generate three networks of 1000 nodes each using Random Network Model, Small World Network Model, Preferential Attachment Model and compare their characteristics.
4. Compute different centrality measures to identify top-N nodes and compare their ranks with those obtained by PageRank method.
5. Apply community detection algorithms on a small real-world network (e.g. Karate club) and compare modularity using bar plot. Also plot the communities revealed with different colors.
6. Simulate diffusion trends for different epidemic models and present results using appropriate visuals.

### **Pathway: Image Processing**

### **DSE: Computer Graphics**

**Course Prerequisites :** DSC 03 (Mathematics for Computing - I), DSC 04 (Object Oriented Programming with C++)

### **Course Objective**

This course introduces fundamental concepts of Computer Graphics with focus on modeling, rendering and interaction aspects of computer graphics. The course emphasizes the basic principles needed to design, use and understand computer graphics system.

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Describe Standard raster and vector scan devices as well as Graphical Input and output devices
2. Implement algorithms for drawing basic primitives such as line, circle and ellipse.
3. Implement algorithms for line clipping, polygon clipping and polygon filling.
4. Implement a 3D object representation scheme, carryout 2D and 3D transformation, 3D projections
5. Implement visible surface determination algorithms, Illumination models and surface rendering methods
6. Implement a simple computer animation algorithm

## Syllabus

**Unit 1 Introduction:** Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics. Architecture of Raster and Random scan display devices, input/output devices.

**Unit 2 Drawing and clipping primitives:** Raster scan line, circle and ellipse drawing algorithms, Polygon filling, line clipping and polygon clipping algorithms

**Unit 3 Transformation and Viewing:** 2D and 3D Geometric Transformations, 2D and 3D Viewing transformations (Projections- Parallel and Perspective), Vanishing points.

**Unit 4 Geometric Modeling:** Polygon Mesh Representation, Cubic Polynomial curves (Hermite and Bezier).

**Unit 5 Visible Surface determination and Surface Rendering:** Z-buffer algorithm, List-priority algorithm and area subdivision algorithm for visible surface determination. Illumination and shading models, RGB Color model and Basics of Computer Animation.

## References

1. Hearn, D & Baker, M.P. *Computer Graphics*, 2<sup>nd</sup> edition, Prentice Hall of India, 2009.
2. Foley, J. D., Dam, A.V, Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice in C*, 2<sup>nd</sup> edition, Pearson education, 2002.
3. Rogers, D. F. *Mathematical Elements for Computer Graphics*, 2<sup>nd</sup> edition, McGraw Hill Education, 2017.



## Additional References

- (i) Bhattacharya, S. *Computer Graphics*, Oxford University Press, 2018.
- (ii) Marschner, S., & Shirley, P. *Fundamentals of Computer Graphics*, 4<sup>th</sup> edition CRC Press, 2017.

## Suggested Practical List

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement a midpoint circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to fill a polygon using the Scan line fill algorithm.
6. Write a program to apply various 2D transformations on a 2D object (use homogeneous Coordinates).
7. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
8. Write a program to draw Hermite /Bezier curve.

## Pathway: Image Processing DSE: Digital Image Processing

**Course Prerequisites:** DSC01 (Programming using Python), DSC 03 (Mathematics for Computing - I)

## Course Objective

This course introduces students to the fundamentals of digital image processing, It introduces image processing in the Spatial and Frequency domains including techniques for various image transformations, image enhancement/filtering, image restoration, image compression and segmentation and morphological image processing.

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Understand the fundamentals of Image Processing and its role and importance in a variety of applications.
2. Write programs to read/write and manipulate images for the purpose of enhancement.

3. Understand the need for image transforms and their properties.
4. Understand different causes for image degradation and use various techniques to restore images.
5. Understand the need and techniques for image compression.
6. Perform morphological image processing and image segmentation.
7. Develop an image processing application.

## Syllabus

**Unit 1 Introduction:** Digital Image Fundamentals, Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships between Pixels Types of images.

**Unit 2 Spatial Domain Filtering:** Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothing Spatial Filters-Low pass filters, Order Statistics filters; Sharpening Spatial Filters- Laplacian filter.

**Unit 3 Filtering in Frequency Domain:** The Discrete Fourier Transformation (DFT), Frequency Domain Filtering:-Ideal and Butterworth Low pass and High pass filters

**Unit 4 Image Degradation/Restoration Process:** Noise models, Noise Restoration Filters

**Unit 5 Image Compression:** Fundamentals of Image Compression, Huffman Coding, Run Length Coding

**Unit 6 Morphological Image Processing:** Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

**Unit 7 Image Segmentation:** Point, Line and Edge Detection, Thresholding.

## References

1. Gonzalez, R. C., & Woods, R. E. *Digital Image Processing*, 4<sup>th</sup> edition, Pearson education, 2017 .

## Additional References

- (i) Castleman, K. R. *Digital Image Processing*, 1<sup>st</sup> edition, Pearson education,,2007.
- (ii) Gonzalez, R. C., Woods, R. E., & Eddins, S. *Digital Image Processing using MATLAB*, Pearson education Inc., 2004.

- (iii) Jain, A. K. *Fundamentals of Digital Image Processing*, 1<sup>st</sup> edition, Prentice Hall of India, 1988.

### **Suggested Practical List**

The Practicals are to be conducted using Python.

Become familiar with basic Python libraries for Image Processing like OpenCV, Scikit-Image etc.

1. Perform the following:
  - a. Read and display an image.
  - b. Resize a given image.
  - c. Convert a given color image into a corresponding gray-scale image.
  - d. Convert a given color/gray-scale image into black & white image f. Draw the image profile.
  - e. Separate a given color image into three R G & B planes.
  - f. Create a color image using separate three R, G and B planes.
  - g. Write given 2-D data in an image file.
2. To write and execute image processing programs using point processing method:
  - a. Obtain Negative image
  - b. Obtain Flip image
  - c. Thresholding
  - d. Contrast stretching
3. To write and execute programs for image arithmetic operations:
  - a. Addition of two images
  - b. Subtract one image from other image
  - c. Calculate mean value of image
  - d. Different Brightness by changing mean value
4. To write and execute programs for image logical operations:
  - a. AND operation between two images
  - b. OR operation between two images
  - c. Calculate intersection of two images
  - d. Water Marking using X-OR operation
  - e. NOT operation (Negative image)
5. To write and execute a program for histogram calculation and equalization:
  - a. Using inbuilt function
  - b. Without using inbuilt function

6. To write and execute a program performing the following geometric transformations on an image:
  - a. Translation
  - b. Scaling
  - c. Rotation
  - d. Shrinking
  - e. Zooming
7. To understand various image noise models and to write programs for:
  - a. Image restoration
  - b. Remove Salt and Pepper Noise
  - c. Minimize Gaussian noise
  - d. Median filter and Weiner filter
8. Write and execute programs to remove noise from images using spatial filtering.
  - a. Understand 1-D and 2-D convolution process
  - b. Use 3x3 Mask for low pass filter and high pass filter
9. Write and execute programs for image frequency domain filtering.
  - a. Apply FFT on given image
  - b. Perform low pass and high pass filtering in frequency domain
  - c. Apply IFFT to reconstruct image
10. Write and execute a program for edge detection using different edge detection mask.
11. Write and execute a program for image morphological operations erosion and dilation

**Pathway: Image Processing**  
**DSE: Information and Image Retrieval**

**Course Prerequisites** : Digital Image Processing

**Course Objective**

This course introduces students to the fundamentals of information retrieval extending into image retrieval. It lays the theoretical foundation of various essential concepts related to image searches, together with examples of natural and texture image types. It will provide an insight to content-based image retrieval, understanding of the technologies, and solutions of content-based image retrieval.

**Course Learning Outcomes**

On the successful completion of the course, the student would be able to:

1. Understand the concept of information retrieval and the information retrieval models.
2. Understand the working of Text based and content based image retrieval systems.
3. Identify and evaluate the use of content-based features in indexing and retrieval of various types of media content
4. Extract different visual features from images
5. Understand indexing and the semantics of visual data
6. Understand query specification and evaluate the retrieval

## Syllabus

**Unit 1 Introduction to IR:** An example information retrieval problem, the extended Boolean model versus ranked retrieval, The term vocabulary and postings lists: Tokenization, stop words, Normalization (equivalence classing of terms), Stemming and lemmatization, term weighting model: Inverse document frequency, Tf-idf weighting, Information retrieval system evaluation.

**Unit 2 CBIR and feature extraction:** Image Retrieval: Multimedia Information retrieval , Text Based Image Retrieval (TBIR), Content Based Image Retrieval (CBIR), Hybrid systems. Architecture of a typical CBIR system, Low-level features of an image: Color – color space, color moments, color histogram, color coherence vector (CCV), color correlogram, invariant color features.

Texture – Tamura features, coarseness, contrast, SAR Model, Wavelet transform feature. Shape- Moment invariants, turning angles, Fourier descriptors.

**Unit 3 Similarity measures and Performance evaluation:** Similarity measures used in content-based image retrieval: Minkowski-form distance, Mahalanobis distance, Canberra distance, Earth Mover distance, Quadratic form distance

Performance evaluation used in content-based image retrieval: user Comparison, precision and recall, P-R graph, Average Precision, F-measure, Average Normalized Modified Retrieval Rank (ANMRR)

**Unit 4 CBIR systems:** QBIC: Query by Image Content, VIR, VisualSEEK, WebSEEK, NeTRA, MARS: Multimedia Analysis and Retrieval System, SIMPLicity.

**Unit 5 Content-Based Image Retrieval-Challenges:** Semantic gap: Introduction to semantic gap. Bridging the semantic gap: Relevance feedback, multi-modal fusion. Semantic similarity: WordNet.

“Curse of Dimensionality”: Feature Dimensionality reduction, Methods for dimensionality reduction: Principal Component Analysis (PCA), Fisher Linear Discriminant Analysis (FLDA), Local Fisher Discriminant Analysis (LFDA), Isometric Mapping (ISOMAP), Locally Linear Embedding (LLE), and Locality Preserving Projections (LPP).

## References

1. C. Manning, P. Raghavan, and H. Schutze *Introduction to Information Retrieval* Cambridge University Press, 2009 .
2. Vipin Tyagi *Content-Based Image Retrieval: Ideas, Influences, and Current Trends*, Springer, 2018.

### Suggested Practical List

To be implemented in Python

1. Write a program to compute the edit distance between strings s1 and s2. (Hint. Levenshtein Distance)
2. Write a program to Compute Similarity between two text documents.
3. Write a program for Pre-processing of a Text Document: stop word removal.
4. Consider 3 documents as below:-
  - Doc 1: Ben studies about computers in Computer Lab.
  - Doc 2: Steve teaches at Brown University.
  - Doc 3: Data Scientists work on large datasets.
 Perform search on these documents with the following query: Data Scientists and, calculate  $tf * idf$  for data and Scientists in all the documents.
5. Write a program to find out the similarity between document d1 and d2 (refer question#4 ) using cosine similarity method.
6. Write a program to calculate the color moments, color histogram, color coherence vector (CCV), color correlogram for an image.
7. Write a program to find out the similarity between two images using:-
  - a. Minkowski-form distance
  - b. Mahalanobis distance
  - c. Canberra distance
  - d. Earth Mover distance
  - e. Quadratic form distance
8. Given a confusion matrix

		ACTUAL	
		Negative	Positive
PREDICTION	Negative	60	8
	Positive	22	10

Write a program to find precision and recall, Average Precision, F-measure, Average Normalized Modified Retrieval Rank (ANMRR).

## DSE: Combinatorial Optimisation

---

**Course Prerequisites:** None

### Course Objective

This course is designed to introduce the fundamentals of combinatorial optimization to the students in terms of both theory and applications, so as to equip them to explore the more advanced areas of convex and non-convex optimizations.

### Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Model problems using linear and integer programs
2. Apply polyhedral analysis to develop algorithms for optimization problems
3. Use the concept of duality for design of algorithms

### Syllabus

**Unit 1 Introduction:** Introduction to Combinatorial Optimization Problems, Linear and Integer Programs- LP Formulation, understanding integer programs, computational complexities of IP vs LP, using LP to find optimal or approximate integral solutions, concept of integrality gap.

**Unit 2 Theory of Linear Programming and Algorithmic Perspective to Simplex Method:** standard vs. equational form, basic feasible solutions, convexity and convex polyhedra, correspondence between vertices and basic feasible solutions, geometry of Simplex algorithm, exception handling (unboundedness, degeneracy, infeasibility), Simplex algorithm, avoiding cycles.

**Unit 3 Primal-Dual Algorithms:** interpretation of dual, optimality conditions for primal and dual, weak and strong duality, complementary slackness, primal-dual algorithm for the shortest path problem.

**Unit 4 Network Flows:** linear programming formulations for network flows and bipartite matching, totally unimodular matrices.

### References

1. Matousek & Gartner *Understanding and Using Linear Programming*, Springer, 2007 .
2. Papadimitriou, C.H. & Steiglitz, K. *Combinatorial Optimization: Algorithms and complexity*, New edition, Dover Publications inc. , 2000.

## Additional References

- (i) Bazaraa, M.S., Jarvis, J.J., & Sherali, H.D. *Linear Programming and Network Flows*, 4<sup>th</sup> edition, Wiley, 2010.
- (ii) Korte, B., & Vygen, J. *Combinatorial Optimization*, 6<sup>th</sup> edition, Springer, 2018.

## Tutorials

Tutorials based on Theory

### Pathway: Theoretical Computer Science

### DSE: Advanced Algorithms

---

**Course Prerequisites:** Discrete Structures, Design and Analysis of Algorithms, Combinatorial Optimization

## Course Objective

This course is designed to provide exposure to more sophisticated algorithms for some tractable problems, some advanced topics in algorithms such as NP Completeness and how to handle NP hard problems in practice.

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Understand and develop more sophisticated algorithms using some of the nown design techniques.
2. Identify NP hard problems.
3. Use polynomial time reductions to prove NP hardness of problems.
4. Design approximation algorithms for NP hard problems and find their approximation ratio.

## Syllabus

**Unit 1 More applications of Divide and Conquer, Greedy and Dynamic Programming approaches:** Counting Inversions, Closest pair of points, Integer Multiplication, Huffman Code, Segmented Least Squares etc.

**Unit 2 Network Flows:** Ford Fulkerson algorithm for max flow problem.



**Unit 2 Backtracking:** Constructing All Subsets, Constructing All Permutations, Constructing all paths in a graph.

**Unit 3 Polynomial time reductions via gadgets:** SAT and 3-SAT problems; Reducing 3-SAT to Independent set, Clique and Vertex cover.

**Unit 4 Proving NP completeness:** Circuit satisfiability, 3-SAT, Sequencing Problems, Graph coloring, Subset sum.

**Unit 5 Introduction to Approximation Algorithms:** Definition, Concept of approximation factor, Bounding the optimal solution, concept of tight example.

**Unit 6 Combinatorial Approximation Algorithms:** Set cover, Minimizing makespan, k-center.

**Unit 7 LP based Approximation Algorithms:** Approximation algorithms for Vertex cover/Set cover via LP rounding.

#### References

1. Kleinberg, J., Tardos, E. *Algorithm Design*, 1<sup>st</sup> edition, Pearson, 2013.
2. Vazirani, V. V. *Approximation Algorithms*, 1<sup>st</sup> edition, Springer, 2001.

#### Additional References

- (i) Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4<sup>th</sup> edition, Prentice Hall of India, 2022.
- (ii) Williamson, D. P., Shmoys, D. B. *The Design of Approximation Algorithms*, 1<sup>st</sup> edition, Cambridge University Press, 2011.

#### Tutorials

Tutorials based on Theory

**Pathway: AI and Machine Learning**

**DSE: Deep Learning**

**Course Prerequisites:** DSC03(Mathematics for Computing - I), DSC17 (Machine Learning)

## **Course Objective**

The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

## **Course Learning Outcomes**

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

1. Describe the feed-forward and deep networks.
2. Design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
3. Implement deep neural networks to solve a problem
4. Analyze performance of deep networks.
5. Use pre-trained models to solve a problem.

## **Syllabus**

**Unit 1 Introduction to neural networks:** Artificial neurons, perceptron, computational models of neurons, Structure of neural networks, Multilayer feedforward neural networks (MLFFNN), Backpropagation learning, Empirical risk minimization, bias-variance tradeoff, Regularization, output units: linear, softmax , hidden units: tanh, RELU

**Unit 2 Deep neural networks:** Difficulty of training DNNs, Greedy layerwise training, Optimization for training DNN's, Newer optimization methods for neural networks(AdaGrad, RMSProp, Adam), Regularization methods(dropout, drop connect, batch normalization).

**Unit 3 Convolution neural networks(CNNs):** Introduction to CNN - convolution, pooling, Deep CNNs - LeNet, AlexNet. Training CNNs, weights initialization, batch normalization, hyperparameter optimization, Understanding and visualizing CNNs, Using a pre trained convnet

**Unit 4 Recurrent neural networks (RNNs):** Sequence modeling using RNNs, Backpropagation through time, LongShort Term Memory (LSTM), Bidirectional RNN, Bidirectional LSTM

**Unit 5 Unsupervised deep learning:** Autoencoders, Generative Adversarial Networks.

**Unit 6 Applications:** Computer vision, Speech recognition and NLP.

## References

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT Press Book, 2016.
2. Francois Chollet, *Deep Learning with python, 2<sup>nd</sup> edition*, Meaning Publications Co, 2021.

## Additional References

- (i) Bunduma, N., *Fundamentals of Deep Learning*, 1<sup>st</sup> edition, O'reilly Books, 2017.
- (ii) Heaton, J., *Deep Learning and Neural Networks*, 1<sup>st</sup> edition, Heaton Research Inc., 2015.

## Suggested Practical List

The following practicals are to be conducted using Python.

1. Implement a feed-forward neural networks for classifying movie reviews as positive or negative(using IMDB dataset)
2. Implement a deep-neural feed-forward network for estimating the price of house, given real-estate data(Boston Housing Price)
3. Implement a deep-neural network for classifying news wires by topic (Reuters dataset).
4. Implement CNN for classifying MNIST dataset
5. Create a model for time-series forecasting using RNN/LSTM
6. Implement an auto-encoder

## Pathway: AI and Machine Learning

### DSE: Reinforcement Learning

**Course Prerequisites:** Machine Learning, Artificial Intelligence,

## Course Objective

The objectives of this course are:

1. to prepare students to visualize reinforcement learning problems
2. to introduce students to the concepts based on Markov Decision Process, Dynamic Programming, Monte Carlo methods, and Temporal-Difference learning.
3. recognize current advanced techniques and applications in Reinforcement Learning

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. learn Reinforcement Learning task formulations and the core principles behind Reinforcement Learning.
2. work on problem-solving techniques based on Dynamic Programming, Monte Carlo, and Temporal-Difference.
3. implement in code common algorithms following code standards and libraries used in Reinforcement Learning.
4. learn the policy gradient methods from vanilla to relatively complex cases.

## Syllabus

**Unit 1 Introduction:** Historical perspective of Reinforcement Learning (RL), Basics of RL: definition, how reinforcement learning happens, examples, terminology, notation, and assumptions, Elements of RL: polices, value function, reward Functions and Bellman Equation, different techniques for solving RL problem, Code Standards and Libraries used in RL using Python/Keras/TensorFlow/MATLAB.

**Unit 2 Markov Decision Process (MDP) and Dynamic Programming (DP):** Markov property, Introduction to Markov decision process (MDP), creating MDPs, goals and rewards, returns and episodes, optimality of value functions and policies, Bellman optimality equations. Overview of dynamic programming for MDP, principle of optimality, iterative policy evaluation, Policy Improvement, policy iteration, value iteration, generalized policy iteration, Asynchronous DP, Efficiency of DP.

**Unit 3 Monte Carlo (MC) Methods:** Monte Carlo methods (First visit and every visit Monte Carlo), Monte Carlo control, On policy and off policy learning, Importance sampling.

**Unit 4 Temporal Difference (TD) Learning:** Temporal-Difference learning methods - TD (0), SARSA, Q-Learning and their variants. Markov reward process (MRP), Overview of TD (1) and TD( $\lambda$ ).

**Unit 5 Approximation Methods and Policy Gradient:** Function approximation methods (Gradient MC and Semi-gradient TD (0) algorithms), Eligibility traces, After-states, Least squares TD. Policy Approximation and its advantages, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods, an introduction to Deep Reinforcement Learning

## References

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction* 2<sup>nd</sup> Edition, MIT Press, 2018.

2. Enes Bilgin *Mastering Reinforcement Learning with Python: Build next-generation, self-learning models using reinforcement learning techniques and best practices*, 1st edition, Packt Publishing, 2020.

### **Additional References**

- (i) Phil Winder *Reinforcement Learning: Industrial Applications of Intelligent Agents*, O'Reilly Media, 2020.
- (ii) Alexander Zai, Brandon Brown *Deep Reinforcement Learning in Action*, 1<sup>st</sup> edition, Manning Publications, 2020.

### **Suggested Practical List**

Implement the following exercises using Python/Keras/TensorFlow/MATLAB.

1. Dynamic Programming Policy Evaluation algorithm.
2. Dynamic Programming Policy Iteration algorithm.
3. Dynamic Programming Value Iteration algorithm.
4. Monte Carlo Prediction
5. Off-Policy Monte Carlo Control with Importance Sampling
6. SARSA On policy TD learning algorithm
7. Q-learning OFF policy TD learning algorithm.
8. Policy Gradient REINFORCE algorithm
9. Policy Gradient Actor-Critic method algorithm

*\*For exercises 1 to 7, consider the following environments for testing: GridWorld, Blackjack, WindyGridWorld*

*\*For exercises 8 onward, consider the following environments for testing: CartPole, CartPoleRaw*

**Pathway: AI and Machine Learning**

**DSE: Natural Language Processing**

**Course Prerequisites:** Probability and Statistics, Machine Learning, Deep Learning

### **Course Objective**

The objectives of this course are:

1. To introduce foundational understanding in natural language

2. To understand the principles and methods of statistical natural language processing
3. To develop an in-depth understanding of the algorithms available for the processing and analysis of natural languages
4. To perform statistical analysis of textual data and find useful patterns from the data

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Grasp the significance of natural language processing in solving real-world problems
2. Preprocess and Analyze text using mathematical techniques.
3. Apply machine learning techniques used in NLP - HMM, RNN
4. Understand approaches to syntax and semantics analysis in NLP
5. Gain practical experience of using NLP toolkits

### **Syllabus**

**Unit 1 Introduction and Basic Text Processing:** Knowledge in Speech and Language Processing, The problem of ambiguity, Typical NL Tasks, Tokenization, Stemming, Lemmatization, Stop-word removal

**Unit 2 Formal Language Modeling:** Regular Expressions, Text Normalization, and Edit Distance, Unigrams, Bigrams, N-grams, N-gram Language Models, Smoothing and Entropy

**Unit 3 Sequence Labeling for Parts of Speech Tagging:** Part-of-Speech Tagging, Named Entities and Named Entity Tagging/Recognition, Hidden Markov Model (Forward and Viterbi algorithms and EM training)

**Unit 4 Vector Semantics and Embedding:** Lexical Semantics, Vector Semantics, Words and Vectors, TF-IDF: Weighing terms in the vector and its applications, Learning Word Embeddings - Word2vec and Gensim, Vector Space Models

**Unit 5 Applications of Text Mining:** Text classification, Sentiment Analysis

**Unit 6 Deep Learning Models for NLP:** Feedforward Neural Networks, Recurrent Neural Networks, and LSTM

### **References**

1. Daniel Jurafsky and James H. Martin *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 3<sup>rd</sup> edition, Pearson, 2022.
2. Christopher D. Manning and Hinrich Schütze *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

3. Steven Bird, Ewan Klein, and Edward Loper *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*, 1<sup>st</sup> edition, O'Reilly Media, 2009.
4. Yoav Goldberg *A Primer on Neural Network Models for Natural Language Processing*, 2022.

### Suggested Practical List

#### Python Packages like Scikit (SKLearn), NLTK, spaCy, gensim, PyTorch, transformers (HuggingFace) etc. may be used for programming

1. Prepare/Pre-process a text corpus to make it more usable for NLP tasks using tokenization, filtration of stop words, removal of punctuation, stemming and lemmatization.
2. List the most common words (with their frequency) in a given text excluding stopwords.
3. Extract the usernames from the email addresses present in a given text. .
4. Perform POS tagging in a given text file. Extract all the nouns present in the text. Create and print a dictionary with frequency of parts of speech present in the document. Find the similarity between any two text documents
5. Perform dependency analysis of a text file and print the root word of every sentence.
6. Create the TF-IDF (Term Frequency -Inverse Document Frequency) Matrix for the given set of text documents
7. Extract all bigrams , trigrams using ngrams of nltk library
8. Identify and print the named entities using Name Entity Recognition (NER) for a collection of news headlines.
9. Find the latent topics in a document using any LDA and display top 5 terms that contribute to each topic along with their strength. Also visualize the distribution of terms contributing to the topics.
10. Classify movie reviews as positive or negative from the IMDB movie dataset of 50K movie reviews. (Link for dataset:  
<https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews> )
11. Build and train a text classifier for the given data (using textbob or simpletransformers library)
12. Generate text using a character-based RNN using an appropriate dataset. Given a sequence of characters from a given data (eg "Shakespear"), train a model to predict the next character in the sequence ("e").

### Pathway: Cyber Security

## **DSE: Blockchain and its Applications**

**Prerequisites:** Knowledge of some Programming language and databases

### **Course Objective**

This course covers the basic concepts behind blockchain and presents Bitcoin and other cryptocurrencies as the motivation for blockchain technologies. It provides a substantive discussion about different technologies behind blockchain and cryptocurrencies.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. understand the applications of blockchain in different domains
2. learn the practical applications of cryptocurrency such as Bitcoin and Ethereum
3. understand basic technologies like cryptographic hash functions, blocks, merkel trees, elliptic curve cryptography and digital signatures.
4. to have knowledge of decentralized consensus algorithms like proof of work, proof of stack, proof of capacity etc.
5. to learn how to record transactions in blockchain, computing bitcoin address etc.
6. to learn about smart contracts and their applications
7. to learn about permissioned and permission less blockchain and hyperledgers.
8. to gain knowledge of real world aspects of Bitcoin, such as wallets and mining techniques. with the Bitcoin network.

### **Syllabus**

**Unit 1 Introduction:** History of money, Digital Currencies, Ledgers, Cryptography, Centralized and Decentralized systems, peer to peer systems, the purpose of Blockchain, types of blockchain (public, private and semi-private blockchain), application of blockchain (in government, healthcare, real estate, voting, insurance, non-fungible tokens, metaverse, Web 3.0).

**Unit 2 Cryptocurrency and Design:** Concept of cryptocurrency, History of Bitcoin, concept of mining, challenges of blockchain/bitcoin design (performance, scalability, efficiency, security, governance, public policy and legal framework).

**Unit 3 Blockchain Technology:** Properties of hash functions, Cryptographic hash functions, hashes (as names, references and commitments), Blocks, Block Headers, Merkle Trees, chain forks, Asymmetric Cryptography, Digital signatures.

**Unit 4 Decentralized Network Consensus:** Introduction to decentralized networks, Native Currency, consensus, proof of work (PoW), proof of stake (PoS), proof of capacity (PoC),



proof of burn (PoB), Practical Byzantine Fault Tolerance (pBFT), Proof of Elapsed Time (PoET).

**Unit 5 Permissioned and Permissionless blockchain:** Blockchain systems vs. traditional databases, introduction to permissioned/permissionless blockchains and their applications, Advantages and disadvantages, Solidity.

**Unit 6 Blockchain and Money Transactions:** Satoshi and Bitcoin, Recording of transactions in blockchain, transaction inputs, outputs and format, Bitcoin address.

**Unit 7 Smart contracts (Ethereum and other currencies):** Overview of smart contracts, tokens and Ethereum as a platform for smart contracts, blockchain technology as regulatory authority.

## References

1. Imran Bashir *Mastering blockchain Distributed ledger technology, decentralization, and smart contracts explained*, 2<sup>nd</sup> edition, Packt Publication, 2018.
2. Lorne Lantz and Daniel Cawrey *Mastering Blockchain Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications*, 1<sup>st</sup> edition, O'Reilly Publication, 2020.
3. Chris Dannen *Introducing Ethereum and Solidity Foundations of Cryptocurrency and Blockchain Programming for Beginners*, 1<sup>st</sup> edition, Apress Publication, 2017.

## Additional Reference

- (i) Daniel Drescher *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 1<sup>st</sup> edition, Apress Publication, 2017.

## Suggested Practical List

1. Using SHA256, obtain the message digest of string "Blockchain Developer".
2. Write a program to encrypt and decrypt the message "Hello World" using SHA256.
3. Implement a simple chain of 5 nodes using linked list.
4. Implement RSA cryptographic algorithm.
5. Create a simple blockchain using Proof of Work (PoW).
6. Demonstrate sending of a digitally signed document.
7. Create a hash table that has '8' number of buckets and insert the keys 20, 1, 7, 15, 25, 16, 8 and 20 using hash function  $h(k) = k \bmod 3$ .
8. Create a blockchain having 5 nodes and print the hash values of each block.
9. Create a blockchain having 5 nodes and check its validity.
10. Use flask to deploy blockchain.

## **Pathway: (Other DSEs)**

### **DSE: Microprocessor**

**Course Prerequisites:** DSC02 (Computer System Architecture)

#### **Course Objective**

This course introduces internal architecture, programming models of Intel Microprocessors (8086 - Pentium) and assembly language programming. Students will also learn interfacing of memory and I/O devices with microprocessors.

#### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Describe the internal architecture of Intel microprocessors.
2. Define and implement interfaces between the microprocessor and the devices.
3. Write assembly language programs.

#### **Syllabus**

**Unit 1 Microprocessor architecture:** Internal architecture, Programming Model, Addressing modes, Data movement instructions.

**Unit 2 Microprocessor programming:** Register Organization, instruction formats, Program control instructions, assembly language.

**Unit 3 Interfacing:** Bus timings, Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, timer, Interrupt controller, DMA controller, video controllers, communication interfaces.

**Unit 4 Data transfer schemes:** Synchronous data transfer, asynchronous data transfer, interrupt driven data transfer, DMA mode data transfer.

**Unit 5 Microprocessor controllers:** I/O controllers, interrupt controller, DMA controller, USART controller.

**Unit 6 Advanced microprocessor architecture:** CISC architecture, RISC architecture, superscalar architecture, multicore architecture.

## References

1. Brey, B.B. *The Intel Microprocessors: Architecture, Programming and Interfacing*, 8<sup>th</sup> edition, Pearson education, 2009.
2. Triebel, W.A., & Singh, A. *The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware and Applications*, 4<sup>th</sup> edition, Pearson education, 2002.

## Additional References

- (i) Ramesh S Gaonkar *Microprocessor architecture, programming, and applications with the 8085*, 6<sup>th</sup> edition, Penram International Publishing, 2013.

## Suggested Practical List

### ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program to print 'Hello World'.
2. Write a program to print two strings on two different lines.
3. Write a program to take a single digit number from the user and print that number on the console.
4. Write a program to compare two single digit numbers and check if they are equal or not.
5. Write a program for 8-bit addition of two single digit numbers. Show the result after ASCII adjust.
6. Write a program for 16-bit addition of two double digit numbers. Show the result after ASCII adjust.
7. Write a program for 16-bit BCD addition.
8. Write a program for 32-bit BCD addition and subtraction.
9. Write a program for 32-bit Binary addition, subtraction, multiplication and division.
10. Write a program for Binary to ASCII conversion.
11. Write a program for ASCII to Binary conversion.
12. Write a program to take input in an array and print it on the console.
13. Write a program to sort an array using bubble sort.
14. Write a program to perform linear search in an array.
15. Write a program to perform binary search in an array.
16. Write a program to add and subtract two arrays.

**Pathway: (Other DSEs)**

## **DSE: Introduction to Web Programming**

**Course Prerequisites :** None

### **Course Objective**

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple websites using HTML, CSS and Javascript.

### **Course Learning Outcomes**

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

1. Build websites using the elements of HTML.
2. Build dynamic websites using the client side programming techniques with CSS, Javascript and jQuery.
3. Learn to validate client-side data.

### **Syllabus**

**Unit 1 Introduction:** Introduction to internet and web design. Basic concepts of web architecture.

**Unit 2 HTML:** Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames.

**Unit 3 Cascading style sheet (CSS):** Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding properties, margin properties).

**Unit 4 Javascript:** Document object model, data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators, validations.

**Unit 4 jQuery and JSON:** Introduction to jQuery, syntax, selectors, events. JSON file format for storing and transporting data.

### **References**

1. Nixon, R. *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Reilly, 2018.
2. Powell, T.A. *HTML & CSS: The Complete Reference, 5<sup>th</sup> edition*, Tata McGrawHill, 2010.
3. Duckett, J. *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley, 2014.

### **Additional References**

- (i) Minnick, J. *Web Design with HTML5 and CSS3*, 8<sup>th</sup> edition, Cengage Learning, 2015.
- (ii) Boehm, A., & Ruvalcaba, Z. *Munarch's HTML5 and CCS*, 4<sup>th</sup> edition, Mike Murach & Associates, 2018.
- (iii) J. A. Ramalho *Learn Advanced HTML 4.0 with DHTML*, BPB Publications, 2007.
- (iv) Ivan Bayross *Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI*, BPB Publications, 2009.

### **Suggested Practical List**

#### **HTML**

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create an HTML displaying this semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame showing your college's name and logo. Bottom horizontal frame split into two vertical frames. The left frame with hyperlinks to pages related to faculty, courses, student activities, etc. The right frame showing corresponding pages based on the link clicked on the left frame.
1. Create a student registration form using HTML which has the following controls:
  - I. Text Box
  - II. Dropdown box
  - III. Option/radio buttons
  - IV. Check boxes
  - V. Reset and Submit button

#### **CSS**

Create a webpage for your department with drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors, ID, class.

## **Javascript**

1. Create event driven programs for the following:
  - a. Enter a number and on click of a button print its multiplication table.
  - b. Print the largest of three numbers entered by the user.
  - c. Find the factorial of a number entered by the user.
  - d. Enter a list of positive numbers using the prompt terminated by a zero. Find the sum and average of these numbers.
2. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customise the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:
  - a. Roll number is a 7-digit numeric value
  - b. Name should be an alphabetical value(String)
  - c. Non-empty and valid fields like DOB

## **jQuery and JSON**

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery
4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

**DSE: Research Methodology  
(Under Preparation)**

**DSE: Research Methodology  
(Under Preparation)**

**Pathway: (Other DSEs)**

**DSE: Distributed Algorithms**

**Course Prerequisites : None**

## Course Prerequisites

C++, Data Structures and Algorithms-1, Operating Systems, Design and Analysis of Algorithms, Theory of Computation

## Course Objective

The course introduces the students to distributed algorithms in synchronous and asynchronous network models. The course would give the students hands-on practice to write programs for distributed algorithms using Remote Procedure Call (RPC) or Message Passing Interface (MPI)

## Course Learning Outcomes

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

1. Describe Network Models for distributed Algorithms
2. Develop elementary synchronous distributed algorithms
3. Develop elementary asynchronous distributed algorithms

## Syllabus

**Unit 1 System Model/Network Models:** Synchronous Network Model, Asynchronous System Model, Asynchronous Network Model

**Unit 2 Synchronous Network Algorithms:** Distributed problems in Synchronous Networks such as Leader Election in a Synchronous Ring. Algorithms in General Synchronous Networks (for example Leader Election in a General Network, Breadth-First Search, Maximal Independent Set etc). Problems of reaching consensus in a distributed network namely, distributed consensus with link failures coordinated Attack Problem (Deterministic Version and Randomized Version) and distributed consensus with link failures(Stopping failures, Introduction to Byzantine Failures). More Consensus Problems such as the k-Agreement etc.

**Unit 3 Asynchronous Network Algorithms:** Basic Asynchronous Network Algorithms such as Leader Election in a Ring, Leader Election in an Arbitrary Network etc. Logical Time Asynchronous Networks, Adding Logical Time to Asynchronous Algorithms, Applications such as Banking System etc. Basics of Network Resource Allocation (mutual Exclusion, resource allocation etc) and Basics of Asynchronous Networks with Process Failures such as k-Agreement etc.

## References

1. Lynch, N. *Distributed Algorithms*, Morgan Kaufmann Publishers, Inc., 1996.

### **Additional References**

- (i) Garg, V. *Elements of Distributed Computing*, Wiley, 2014.

### **Suggested Practical List**

1. Implement Leader Election in a Synchronous Ring.
2. Implement Leader Election in a General Network (Synchronous Network)
3. Implement Breadth-First Search (Synchronous Network)
4. Implement Maximal Independent Set (Synchronous Network)
5. Implement Leader Election in an Asynchronous Ring.
6. Implement Asynchronous Banking System  
Optional
7. Implement distributed consensus with link failure (Synchronous Network)
8. Implement distributed consensus with Process failure (Synchronous Network)

## **Pathway: (Other DSEs)**

### **DSE: Internet Technologies - Mobile App Designing**

#### **Syllabus**

**Unit 1 Android Systems:** Introduction to Mobile devices and applications, Open Handset Alliance (OHA), Overview of Android OS and architecture, installing android studio. Introduction to Android application components, Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions Activities and intents: understanding activity and its life cycle, Types of intents, intent filter, context, data sharing using intent

**Unit 2 Android User Interface:** Basic android UI, layouts, view and view attributes, button, controls. UI events and event listeners, animations, notifications, progress dialog, Action bar, toolbar, menus and pop ups, Tab based UI, Fragment, Types of Fragment, Fragment Lifecycle, communication between fragment and activity

**Unit 3 Android Storage and APIs:** Android storage: Using Android Data and Storage APIs, Managing data using SQLite, Sharing Data between Applications with Content Providers  
Android APIs: Multimedia, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, android location based services

**Unit 4 iOS Technology Stack:** Introduction to iOS technology stack: iOS architecture,



Storyboard, features of Xcode, components of iOS SDK. Introduction to swift: data types, variables, control flow and operators, Collections and functions in swift, classes and structures, inheritance, closure and. enumerations

**Unit 5 User interactions:** Controls, gesture organizers, touching views, Core Location and Mapkit, using Google Maps in iOS. Sensors in iOS. Data persistence: Core Data framework for storing persistent data, CRUD operations.

Note: Kotlin will be used for the implementation.

## References

1. Meier Reto and Ian Lake, *Professional Android*, 4<sup>th</sup> edition, Wrox, 2018.
2. Craig Grummitt *iOS Development with Swift*, Manning publications.
3. Rick Boyer *Android 9 Development Cookbook*, Packt Publishing Limited, 2018.

**Department of Computer Science  
University of Delhi**

**Generic Electives (GEs)**

**Non - Computer Science (Hons)**

**Under UGCF 2022**

DRAFT

**Approved in  
UG Committee meeting held on May 17, 2022  
Faculty of Mathematical Sciences meeting held on May 25, 2022**

To obtain a “Minor in Computer Science”, a student must study GE1a, GE 3a - GE6a

1. List of GEs : It is a union of the following List A and List B

- a. List A: Discipline Specific Electives that are available as Generic Electives for students from non-computer science streams provided they satisfy the prerequisites.
- b. List B: [Additional GEs](#)

GEs of Semester I, III, V and VII can be taken in any order so long as the prerequisites are satisfied. Similarly, GEs of Semester II, IV, VI and VIII can be taken in any order so long as the prerequisites are satisfied.

Table of additional GEs

Sem ester	GE -No.	Title	L	T*	P*	Total credit s	Pre-requisites
I	GE1a	<a href="#">Programming using C++</a>	3	0	1	4	None
	GE1b	<a href="#">Programming with Python</a>	3	0	1	4	None
II	GE2a	<a href="#">Data Analysis and Visualization using Python</a>	3	0	1	4	A course in Python
	GE2b	<a href="#">Data Analysis and Visualization using Spreadsheet</a>	3	0	1	4	None
III	GE3a	<a href="#">Database Management Systems</a>					None
	GE3b	<a href="#">Java Programming</a>					None

IV	GE4a	<a href="#">Data Structures using C++</a>	3	0	1	4	A course in C++
	GE4b	<a href="#">Introduction to Web Programming</a>	3	0	1	4	None
V	GE5a	<a href="#">Operating Systems</a>	3	0	1	4	Knowledge of Programming and Data Structures
	GE5b	<a href="#">Advanced Web Programming</a>	3	0	1	4	Knowledge of HTML, CSS
	GE5c	<a href="#">Java Based Web App Development</a>	3	0	1	4	A course in C++/Java
VI	GE6a	<a href="#">Computer Networks</a>	3	0	1	4	Knowledge of C/C++/Java/Python
	GE6b	<a href="#">Internet Technologies: Web App Design and Development</a>	3	0	1	4	A course in C++/Java
	GE6c	<a href="#">Artificial Intelligence</a>	3	0	1	4	Knowledge of any programming language, basics of Algorithms and Data structures
VII	GE7a	<a href="#">Information Security</a>	3	0	1	4	Knowledge of Programming, Database Management Systems, Computer Networks.
	GE7b	<a href="#">Design and Analysis of Algorithms</a>					Knowledge of Programming and Data Structures
	GE7c	<a href="#">Internet Technologies : Mobile App</a>	3	0	1	4	A course in C++/Java

		<a href="#">Design and Development</a>					
VIII	GE8a	<a href="#">Machine Learning</a>	3	1	0	4	A course in probability, statistics, linear algebra and multivariate analysis
	GE8b	<a href="#">Digital Marketing and Social Media Analytics (under preparation)</a>	3	0	1	4	Python and HTML
	GE8c	<a href="#">Introduction to Parallel Programming</a>	3	1	0	4	Knowledge of Computer System Architecture, C++, Knowledge of Basic Data Structures and Algorithms, Operating Systems

**Note:**

1. Wherever C++/Java/Python is a prerequisite, C++/Java/Python of plus 2 level is acceptable.
2. Batch size for Practicals will be (8-10) and Tutorials will be (12-15).

It was recommended to add more electives in Semester VII and VIII in due course of time.

**GE1a: Programming using C++**

**Course Objective**

This course is designed to introduce programming concepts using C++ to students. The course aims to develop structured as well as object-oriented programming skills using C++ programming language. The course also aims to achieve competence amongst its students to develop correct and efficient C++ programs to solve problems in their respective domains.

**Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Write simple programs using built-in data types of C++.
2. Implement arrays and user defined functions in C++.
3. Solve problems in the respective domain using suitable programming constructs in C++.
4. Solve problems in the respective domain using the concepts of object oriented programming in C++.

## Syllabus

**Unit 1 Introduction to C++:** Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++.

**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

**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

## References

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

## Additional References

- (i) Herbert Schildt, *C++: The Complete Reference*, 4<sup>th</sup> edition, McGraw Hill, 2003.
- (ii) A. B. Forouzan, Richard F. Gilberg, *Computer Science: A Structured Approach using C++*, 2<sup>nd</sup> edition, Cengage Learning, 2010.

## Suggested Practical list

1. Write a program to compute the sum of the first n terms of the following series:

$$S = 1 - 2 + 3 - 4 + \dots n$$

The number of terms n is to be taken from the user through the command line. If the command line argument is not found then prompt the user to enter the value of n.

2. Write a program to display the following pattern:

1

22

333

4444

55555

The number of rows  $n$ , is to be taken from the user.

3. Write a program to compute the factors of a given number.
4. Write a menu driven program to perform the following operations on an array:
  - a. Find the minimum, maximum and average of the array elements
  - b. Search an element in the array using linear and binary search
5. Write a menu driven program to perform the following operations on a string:
  - a. Calculate length of the string
  - b. Check whether the first character of every word in the string is in uppercase or not
  - c. Reverse the string
6. Create a class Triangle. Include overloaded functions for calculating the area of a triangle.
7. Create a template class TwoDim which contains  $x$  and  $y$  coordinates. Define default constructor, parameterized constructor and void print() function to print the co-ordinates. Now reuse this class in ThreeDim adding a new dimension as  $z$ . Define the constructors and void print() in the subclass. Implement main() to show runtime polymorphism.

## **GE1b: Programming with Python**

### **Course Objective**

The course is designed to introduce programming concepts using Python to students. The course aims to develop structured as well as object-oriented programming skills using Python. The course also aims to achieve competence amongst its students to develop correct and efficient Python programs to solve problems in their respective domains.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Write simple programs using built-in data structures in Python.
2. Implement arrays and user defined functions in Python.
3. Solve problems in the respective domain using suitable programming constructs in Python.
4. Solve problems in the respective domain using the concepts of object oriented programming in Python.

### **Syllabus**

**Unit 1 Introduction to Programming:** Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

**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.

**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.

**Unit 4 File and exception handling:** File handling through libraries; Errors and exception handling.

### **References**



1. Taneja, S., Kumar, N., *Python Programming- A modular Approach*, Pearson Education India, 2018.
2. Balaguruswamy E., *Introduction to Computing and Problem Solving using Python*, 2nd edition, McGraw Hill Education, 2018.

### Additional References

- (i) Brown, Martin C., *Python: The Complete Reference*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.
- (ii) Guttag, J.V. *Introduction to computation and programming using Python*, 2<sup>nd</sup> edition, MIT Press, 2016.

### Suggested Practical List

1. WAP to find the roots of a quadratic equation.
2. WAP to accept a number 'n' and
  - a. Check if 'n' is prime
  - b. Generate all prime numbers till 'n'
  - c. Generate first 'n' prime numbers
  - d. This program may be done using functions.
3. WAP to create a pyramid of the character '\*' and a reverse pyramid

```

*
***
*****
*****
*****

```

```

*****
*****
*****
***
*

```

4. WAP that accepts a character and performs the following:
  - a. print whether the character is a letter or numeric digit or a special character
  - b. if the character is a letter, print whether the letter is uppercase or lowercase
  - c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. WAP to perform the following operations on a string
  - a. Find the frequency of a character in a string.
  - b. Replace a character by another character in a string.
  - c. Remove the first occurrence of a character from a string.
  - d. Remove all occurrences of a character from a string.
6. WAP to swap the first n characters of two strings.
7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.
8. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:
  - a. 'for' loop
  - b. list comprehension
9. WAP to read a file and
  - a. Print the total number of characters, words and lines in the file.
  - b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
  - c. Print the words in reverse order.
  - d. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.
10. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
11. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:
  - a. Print half the values of the tuple in one line and the other half in the next line.
  - b. Print another tuple whose values are even numbers in the given tuple.

- c. Concatenate a tuple  $t_2=(11,13,15)$  with  $t_1$ .
  - d. Return maximum and minimum value from this tuple
12. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

## **GE2a: Data Analysis and Visualization using Python**

### **Course Objective**

This course is designed to introduce the students to real-world data analysis problems, their analysis and interpretation of results in the field of exploratory data science using Python.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Apply descriptive statistics to obtain a deterministic view of data
2. Perform data handling using Numpy arrays
3. Load, clean, transform, merge and reshape data using Pandas
4. Visualize data using Pandas and matplotlib libraries

### **Syllabus**

**Unit 1 Introduction to basic statistics and analysis:** Fundamentals of Data Analysis, Statistical foundations for Data Analysis, Types of data, Descriptive Statistics, Correlation and covariance, Linear Regression, Statistical Hypothesis Generation and Testing, Python Libraries: NumPy, Pandas, Matplotlib

**Unit 2 Array manipulation using Numpy:** Numpy array: Creating numpy arrays, various data types of numpy arrays, indexing and slicing, swapping axes, transposing arrays, data processing using Numpy arrays

**Unit 3 Data Manipulation using Pandas:** Data Structures in Pandas: Series, DataFrame, Index objects, Loading data into Pandas data frame, Working with DataFrames: Arithmetics,

Statistics, Binning, Indexing, Reindexing, Filtering, Handling missing data, Hierarchical indexing, Data wrangling: Data cleaning, transforming, merging and reshaping

**Unit 4 Plotting and Visualization:** Using matplotlib to plot data: figures, subplots, markings, color and line styles, labels and legends, Plotting functions in Pandas: Line, bar, Scatter plots, histograms, stacked bars, Heatmap

### References

1. McKinney W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython*. 2<sup>nd</sup> edition. O'Reilly Media, 2018..
2. Molin S. *Hands-On Data Analysis with Pandas*, Packt Publishing, 2019.
3. Gupta S.C., Kapoor V.K., *Fundamentals of Mathematical Statistics*, Sultan Chand & Sons, 2020.

### Additional References

- (i) Chen D. Y, *Pandas for Everyone: Python Data Analysis*, Pearson, 2018.
- (ii) Miller J.D. *Statistics for Data Science*, Packt Publishing, 2017.

### Suggested Practical List

Use data set of your choice from Open Data Portal ([https:// data.gov.in/](https://data.gov.in/), UCI repository) or load from scikit, seaborn library for the following exercises to practice the concepts learnt.

1. Load a Pandas dataframe with a selected dataset. Identify and count the missing values in a dataframe. Clean the data after removing noise as follows
  - a. Drop duplicate rows.
  - b. Detect the outliers and remove the rows having outliers
  - c. Identify the most correlated positively correlated attributes and negatively correlated attributes
2. Import iris data using sklearn library or (Download IRIS data from: <https://archive.ics.uci.edu/ml/datasets/iris> or import it from sklearn.datasets)
  - a. Compute mean, mode, median, standard deviation, confidence interval and standard error for each feature

- b. Compute correlation coefficients between each pair of features and plot heatmap
  - c. Find covariance between length of sepal and petal
  - d. Build contingency table for class feature
3. Load Titanic data from sklearn library , plot the following with proper legend and axis labels:
    - a. Plot bar chart to show the frequency of survivors and non-survivors for male and female passengers separately
    - b. Draw a scatter plot for any two selected features
    - c. Compare density distribution for features age and passenger fare
    - d. Use a pair plot to show pairwise bivariate distribution
  4. Using Titanic dataset, do the following
    - a. Find total number of passengers with age less than 30
    - b. Find total fare paid by passengers of first class
    - c. Compare number of survivors of each passenger class

**Project** students are encouraged to work on a good dataset in consultation with their faculty and apply the concepts learned in the course.

## **GE2b: Data Analysis and Visualization using Spreadsheet**

### **Course Objective**

This course is designed to inculcate statistical thinking about data to the students who have studied Mathematics up to Class 10<sup>th</sup> ONLY. It gently introduces the students to basic statistics, and builds gradually to cover advanced functionalities for data analysis in spreadsheets. The objective is to enhance the knowledge of statistics and enable students to make sense of data by analyzing and visualizing it using spreadsheets, interpreting the results and gaining insights.

### **Course Learning Outcomes**

On successful completion of the course, students will be able to:

1. Analyze and visualize data using spreadsheets
2. Apply basic and advanced level statistical functions in spreadsheets
3. Gain practical, hands-on experience of data analysis using spreadsheet

## Syllabus

**Unit 1 Introduction to Basic Statistics:** Fundamentals of Data Analysis, Statistical foundations for Data Analysis, Types of data, Descriptive Statistics, Correlation and covariance, Linear Regression.

**Unit 2 Data Handling:** Spreadsheet concepts, Managing worksheets, Formatting cells, Entering data, Handling operators in formula, Cell referencing and naming of cells and cell ranges, Sorting, Multilayer sorting, Data validation, Find and Replace, Paste special, Filter and advanced filter, Formatting as table, Pivot tables, Formulae vs functions, Cell formulae vs Array formulae.

Mathematical functions, Statistical functions, Logical functions, Date and Time functions, Lookup and reference: Hlookup, and Vlookup, Index and Match functions, Text functions.

What-if-analysis: Goal-seek, Data tables, Scenario manager.

**Unit 3 Data Analysis:** Explore a data model: its content, and its structure, using the Power Pivot add-in. Learning DAX formula language. Create calculated fields and calculated measure for each cell, filter context for calculation, and explore several advanced DAX functions.

Cube formulas to retrieve data from data model.

**Unit 4 Data Visualization:** Different types of charts including Pivot charts: Column, Line, Pie, Bar, Scatter charts. Fine tuning of charts: Chart Elements, Chart Styles, Chart Filters, Box Plot.

## References

1. Gupta, S.P., *Elementary Statistical Methods*, Sultan Chand and Sons, New Delhi, 2017.
2. Goldmeier, J., *Advanced Excel Essentials*, Apress, 2014.
3. Slager, D., *Essential Excel 2016: A Step-by-Step Guide*, Apress, 2016.
4. Valerie M. Sue and Matthew T. Griffin, *Data Visualization and Presentation with Microsoft Office*, SAGE, 2016.
5. Schmuller, J., *Statistical Analysis with Excel for Dummies*, 4<sup>th</sup> edition., Wiley India Pvt Ltd., 2020.

## Suggested Practical List

1. In a meeting of a marketing department of an organization it has been decided that price of selling an item is fixed at Rs. 40. It was resolved to increase the selling of more items and getting the profit of Rs. 50000/-. Use Goal Seek to find out how many items you will have to sell to meet your profit figure.

2. Create worksheet related to crop production of various crops in Indian states in last five years (wheat, rice, pulses, soya-bean, and cane-sugar etc).
  - i) Make a bar chart
  - ii) Make a pie chart
  - iii) Make a box plot
3. Study and perform the various DAX functions to analyse the data.
4. Create workbook related to sales of Business Company having various product in last four quarters for 10 sales persons.
  - i) Make a line graph to show the growth/decline in the sales
  - ii) Show the graph of each sales person sales
  - iii) Find the two sales persons done in last 2 quarters
  - iv) Find the sales persons consistent in last four quarters
  - v) Find the most popular product of the company and the current popular product of company.
5. Create a Pivot-table showing the Customer Names who placed orders with GSS during 2019-2022. For each customer, also show the total number of orders, Total Sales, and Total Profit. Add a Slicer or a Filter that can be used to show the information specifically for each Customer Segment. Use information from the Pivot-table to answer the following questions (Hint: Filter and sort the data in the Pivot-table to locate the answer):
  - i) Which small business customer had the highest sales?
  - ii) Which corporate customer placed the greatest number of orders in 2019-2022? How many orders were placed by the corporate customer?
  - iii) Which consumer customer was the most profitable one?
  - iv) What is the sales figure of the least profitable home office customer?
6. Consider the following worksheet: (enter 5 records)

<b>FULL NAME</b>	<b>GRADE 1/2/3</b>	<b>BASIC SALARY</b>	<b>HRA</b>	<b>PF</b>	<b>GROSS</b>	<b>NET</b>	<b>VA</b>	<b>VA&gt;HRA</b>

HRA is calculated as follows:

Grade	HRA (% of basic)
1	40%
2	35%
3	30%

PF is 8% for all grades

VA is 15000, 10000, 7000 for Grades 1, 2 and 3.

Gross=Basic + HRA+VA

Net=Gross - PF

- i) Find max, min and average salary of employees in respective Grade.
  - ii) Count no. of people where VA>HRA
  - iii) Find out most frequently occurring grade.
  - iv) Extract records where employee name starts with "A" has HRA>10000
  - v) Print Grade wise report of all employees with subtotals of net salary and also grand totals.
  - vi) Use subtotal command.
  - vii) Extract records where Grade is 1 or 2 and salary is between 10000 and 20000 both inclusive.
7. Create workbook related to sales of Business Company having various product in last ten quarters for 20 sales persons. Perform the following on workbook:
- i) Create and modify a Pivot-table
  - ii) Apply Pivot-table styles and formatting
  - iii) Filter a Pivot-table
  - iv) Insert a slicer to filter a Pivot-table
  - v) Create a Pivot Chart
8. Create a PivotTable showing Total Sales breakdown by Region, Product Category, and Product Sub-Category. Use information from the PivotTable to answer the following questions:
- i) What was the Total Sales figure included in this data set?
  - ii) Which Product Category had the highest sales?
  - iii) Which Region had the lowest sales?
  - iv) What was the Total Sales of Appliances in Delhi?







On successful completion of the course, students will be able to:

1. Use relational database management software to create and manipulate the database.
2. Create conceptual data models using entity relationship diagrams for modeling real-life situations and map it to corresponding relational database schema.
3. Use the concept of functional dependencies to remove redundancy and update anomalies.
4. Apply normalization theory to get a normalized database scheme to get anomalies free database.
5. Write queries in relational algebra.
6. Implement relational databases and formulate queries for data retrieval and data update problems using SQL .
7. Learn the importance of index structures and concurrent execution of transactions in database systems.

## **Syllabus**

**Unit 1 Introduction to Database:** Database, characteristics of database approach, data models, database management system, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.

**Unit 2 Entity Relationship Modeling:** Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram as conceptual data model.

**Unit 3 Relational Data Model:** Data anomalies, Relational Data Model - Characteristics of a relation, schema-instance distinction, types of keys, relational integrity constraints. Relational algebra operators like selection, projection, cartesian product, join and write simple queries using them.

**Unit 4 Structured Query Language (SQL):** DDL to create database and tables, table constraints, DML, Querying in SQL to retrieve data from the database, aggregation functions group by and having clauses, generate and query views.

**Unit 5 Database Design:** Mapping an Entity Relationship diagram to corresponding relational database scheme, functional dependencies and Normal forms, 1NF, 2NF, and 3NF decompositions and desirable properties of them.

**Unit 6 Basics of File indexing and introduction to Transaction Processing:** Need of file indexes, types of indexes, introduction to file organizations, basics of concurrent execution of transactions.

### References

1. Elmasri, R., Navathe, B. S., *Fundamentals of Database Systems*, 7<sup>th</sup> Edition, Pearson Education, 2016.
2. Murach J., *Murach's MySQL*, 3<sup>th</sup> Edition, Pearson, 2019.

### Additional References

- (i) Connolly T. M., Begg C. E. *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6<sup>th</sup> edition, Pearson, 2019.
- (ii) Ramakrishnan R., Gehrke J. *Database Management Systems*, 3<sup>rd</sup> Edition, McGraw-Hill, 2014.
- (iii) Silberschatz A., Korth H.F., Sudarshan S. *Database System Concepts*, 7<sup>th</sup> Edition, McGraw Hill, 2019.

### Suggested Practical List

Create and use the following student-course database schema for a college to answer the given queries using the standalone SQL editor.

STUDENT	<u>Roll No</u>	StudentName	CourseID	DOB
		Char(6)	Varchar(20)	Varchar(10)

COURSE	<u>CID</u>	CourseName	Course Type	Teacher-in-charge	TotalSeats	Duration
		Char(6)	Varchar (20)	Char (8)	Varchar (15)	Unsigned int

ADMISSION	<u>Roll No</u>	<u>CID</u>	DateOfAdmission
		Char(6)	Char(6)

Here Rollno (ADMISSION) and SID (ADMISSION) are foreign keys. Note that course type may have two values viz. Fulltime and Parttime and a student may enroll in any number of courses

1. Retrieve names of students enrolled in any course.
2. Retrieve names of students enrolled in at least one part time course.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'
6. Find course details with more than N students enrolled where N is to be input by the user
7. Update student table for modifying a student name.
8. Find course names in which more than five students have enrolled
9. Find the name of youngest student enrolled in course 'BSc(P)CS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two popular part time courses (on the basis of enrolled students)
12. Find the student names who are admitted to full time courses only.
13. Find course names in which more than 30 students took admission
14. Find names of all students who took admission to any course and course names in which at least one student has enrolled
15. Find course names such that its teacher-in-charge has a name with 'Gupta' in it and the course is full time.
16. Find the course names in which the number of enrolled students is only 10% of its total seats.
17. Display the vacant seats for each course
18. Increment Total Seats of each course by 10%
19. Add enrollment fees paid ('yes'/'No') field in the enrollment table.
20. Update date of admission of all the courses by 1 year.
21. Create a view to keep track of course names with the total number of students enrolled in it.
22. Count the number of courses with more than 5 students enrolled for each type of course.
23. Add column Mobile number in student table with default value '9999999999'
24. Find the total number of students whose age is > 18 years.
25. Find names of students who are born in 2001 and are admitted to at least one part time course.
26. Count all courses having 'science' in the name and starting with the word 'BSc'.

**Students are also encouraged to implement the database given in the textbook and do the related queries.**

### **GE3b : Java Programming**

#### **Course Objective**

This course is designed to develop understanding of object-oriented programming concepts like Classes, Objects, Inheritance and Polymorphism using Java. The course provides

understanding of multithreading and exception handling in Java. It also introduces how to create Java applications with graphical user interface (GUI).

### **Course Learning Outcomes**

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

1. Understand the object-oriented concepts – Classes, Objects, Inheritance, Polymorphism– for problem solving.
2. Create and handle multithreading.
3. Handle program exceptions.
4. Handle input/output through files.
5. Create Java applications with graphical user interface (GUI).

### **Syllabus**

**Unit I Introductory Concepts:** program, identifiers, variables, constants, primitive data types, expressions, Naming Conventions, Type casting, operators, control statements, structured data types, arrays, functions.

**Unit II Object Oriented Concepts:** Abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, Anonymous block, Static Data members, overloading and overriding, Usage of super and this keyword, Abstract classes, Interfaces and Packages, Access modifiers, Object class

**Unit III Multithreading:** Creating Threads, Thread Priority, Blocked States, Extending Thread Class, Runnable Interface, Starting Threads, Thread Synchronization, Sync Code Block, Overriding Synced Methods, Thread Communication, wait, notify and notify all.

**Unit IV Introduction to Exception handling:** Exception and Error, Throw, try and catch Blocks, Exception handlers, java.lang Exceptions, Built-InExceptions.

**Unit V Introduction to File Handling:** Byte Stream, Character Stream, File I/O Basics, File Operations, Serialization.

### **References**

1. James Gosling, Bill Joy, Guy L. Steele Jr, Gilad Bracha, Alex Buckley, *The Java Language Specification, Java SE 7<sup>th</sup> edition*, Addison-Wesley, 2013.

2. Herbert Schildt, *Java: The Complete Reference*, 10<sup>th</sup> edition, McGraw-Hill Education, 2018.
3. Cay S. Horstmann, *Core Java - Vol. I – Fundamentals*, 10<sup>th</sup> edition, Pearson, 2017.
4. Richard Johnson, *An Introduction to Java Programming and Object-Oriented Application Development*, Thomson Learning, 2006.
5. Kathy Sierra and Bert Bates, *Head First Java*, 3<sup>rd</sup> edition, O'Reilly, 2022.

### **Suggested Practical List**

1. Create a java program to implement stack and queue concept.
2. Write a program to take input from command line arguments.
3. Write a java program to show static and dynamic polymorphism.
4. Write a java program to show multiple inheritance using interfaces.
5. Write a program in java to show the chaining of execution of construction.
6. Write a java program to show multithreaded producer and consumer application.
7. write a program in java to synchronize the multithreaded application
8. Create a customized exception and also make use of all the exception keywords.
9. Write a program to show different ways to get input from user
10. Design a form using AWT components and Frame container.

### **GE4a : Data Structures using C++**

#### **Course Objective**

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, trees to solve problems. C++ is chosen as the language to understand implementation of these data structures.

## Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Compare two functions for their rates of growth.
2. Understand abstract specification of data-structures and their implementation.
3. Compute time and space complexity of operations on a data-structure.
4. Identify the appropriate data structure(s) for a given application and understand the trade-offs involved in terms of time and space complexity.
5. Apply recursive techniques to solve problems.

## Syllabus

**Unit 1 Growth of Functions, Recurrence Relations:** Functions used in analysis, asymptotic notations, asymptotic analysis, solving recurrences using recursion tree, Master Theorem.

**Unit 2 Arrays, Linked Lists, Stacks, Queues, Deques:** Arrays: array operations, applications, sorting, two-dimensional arrays, dynamic allocation of arrays; Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: stack as an ADT, implementing stacks using arrays, implementing stacks using linked lists, applications of stacks; Queues: queue as an ADT, implementing queues using arrays, implementing queues using linked lists, double-ended queue as an ADT. Time complexity analysis of operations on all data structures.

**Unit 3 Sorting:** Insertion Sort, Count Sort and their complexity analysis.

**Unit 4 Recursion:** Recursive functions, linear recursion, binary recursion.

**Unit 5 Trees, Binary Trees:** Trees: definition and properties, binary trees: definition and properties, traversal of binary trees and their time complexity analysis.

**Unit 6 Binary Search Trees, Balanced Search Trees:** Binary Search Trees: insert, delete (by copying), search operations, time complexity analysis of these operations; Balanced Search Trees and (2,4) Trees: motivation and introduction.

**Unit 7 Binary Heap, Priority Queue:** Binary Heaps: motivation and introduction, application of heaps - Priority Queues.

## References

1. Goodrich, M.T, Tamassia, R., & Mount, D., *Data Structures and Algorithms Analysis in C++*, 2<sup>nd</sup> edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C., *Introduction to Algorithms*. 4<sup>th</sup> edition. Prentice Hall of India. 2022.



3. Drozdek, A., *Data Structures and Algorithms in C++*, 4<sup>th</sup> edition, Cengage Learning, 2012.

### **Additional References**

- (i) Sahni, S. *Data Structures, Algorithms and applications in C++*, 2<sup>nd</sup> edition. Universities Press, 2011.
- (ii) Langsam Y., Augenstein, M. J., & Tanenbaum, A. M. *Data Structures Using C and C++*, 2<sup>nd</sup> edition, Pearson, 2009.

### **Suggested Practical List**

1. Perform matrix addition and multiplication.
2. Implement following recursive functions:
  - a. Factorial of a number
  - b. N<sup>th</sup> fibonacci number
  - c. Power function:  $x^y$
3. Implement singly linked lists.
3. Implement doubly linked lists.
4. Implement circular linked lists.
5. Implement stack data structure and its operations using arrays.
6. Implement stack data structure and its operations using linked lists.
7. Convert Prefix expression to Infix and Postfix expressions, and evaluate.
8. Implement queue data structure and its operations using arrays.
9. Implement queue data structure and its operations using linked lists.
10. Implement Binary Trees and its traversals.

## **GE4b: Introduction to Web Programming**

### **Course Objective**

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple websites using HTML, CSS and Javascript.

### **Course Learning Outcomes**

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

1. Build websites using the elements of HTML.
2. Build dynamic websites using the client side programming techniques with CSS and Javascript.
3. Learn to validate client-side data.

## Syllabus

**Unit 1 Introduction:** Introduction to internet and web design. Basic concepts of web architecture.

**Unit 2 HTML:** Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames.

**Unit 3 Cascading style sheet (CSS):** Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding properties, margin properties).

**Unit 4 Javascript:** Document object model, data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators, validations.

## References

1. Powell, T.A., *HTML & CSS: The Complete Reference*, 5<sup>th</sup> edition, Tata McGrawHill, 2010.
2. Minnick, J. ,*Web Design with HTML5 and CSS3*, 8<sup>th</sup> edition, Cengage Learning, 2015.
3. J. A. Ramalho , *Learn Advanced HTML 4.0 with DHTML*, BPB Publications, 2007.

## Additional References

- (i) Boehm, A., & Ruvalcaba, Z. Murach's, *HTML5 and CCS*, 4<sup>th</sup> edition, Mike Murach & Associates, 2018.
- (ii) Ivan Bayross , *Web Enabled Commercial Application Development Using Html*,

## **Suggested Practical List**

### **HTML**

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create an HTML displaying this semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame showing your college's name and logo. Bottom horizontal frame split into two vertical frames. The left frame with hyperlinks to pages related to faculty, courses, student activities, etc. The right frame showing corresponding pages based on the link clicked on the left frame.
5. Create a student registration form using HTML which has the following controls:
  - a. Text Box
  - b. Dropdown box
  - c. Option/radio buttons
  - d. Check boxes
  - e. Reset and Submit button

### **CSS**

Create a webpage for your department with drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors, ID, class.

### **Javacript**

1. Create event driven programs for the following:
  - a. Enter a number and on click of a button print its multiplication table.
  - b. Print the largest of three numbers entered by the user.

- c. Find the factorial of a number entered by the user
  - d. Enter a list of positive numbers using the prompt terminated by a zero.
  - e. Find the sum and average of these numbers.
2. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customize the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:
- a. Roll number is a 7-digit numeric value
  - b. Name should be an alphabetical value(String)
  - c. Non-empty and valid fields like DOB

## **GE5a: Operating Systems**

### **Course Objective**

The course introduces Operating System and its importance in computer systems. The focus is to explain the common services provided by an operating system like process management, memory (primary, secondary & virtual) management, I/O management, file management. The course talks about the various functional components of the operating and their design.

### **Course Learning Outcomes**

At the successful completion of the course, students will also be able to gain knowledge of different concepts of the operating System and its components. They would learn about shell scripts and would be able to use the system in an efficient manner.

### **Syllabus**

**Unit 1 Introduction:** Operating Systems (OS) definition and its purpose, Multiprogrammed and Time Sharing Systems, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

**Unit 2 Operating System Structures:** OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

**Unit 3 Process Management:** Process Concept, States. Process Control Block, Context Switch, Process scheduling, Schedulers, Overview of threads and Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin.

**Unit 4 Memory Management:** Physical and Logical address space, Swapping Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging, virtual memory: Demand Paging.

**Unit 5 File and Input / Output Device Management:** File Concepts, File Attributes, File Access Methods, Directory Structure: Single-Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories, Magnetic Disks, Solid-State Disks, Magnetic Tapes.

**Unit 6 Shell Scripting:** Shell variables, parameter passing conditional statements, iterative statements, writing and executing shell scripts, utility programs (cut, paste, grep, echo, pipe, filter etc.)

## References

1. Galvin, S. P. B., Gagne, G., *Operating System Concepts*, 9<sup>th</sup> edition, John Wiley Publications, 2016.
2. Das, S., *Unix: Concepts and Applications*, 4<sup>th</sup> Edition, TMH, 2009.

## Additional References

- (i) Dhamdhare, D. M., *Operating Systems: A Concept-based Approach*, 2<sup>nd</sup> edition, Tata McGraw-Hill Education, 2017.
- (ii) Kernighan, B. W., Pike, R., *The Unix Programming Environment*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
- (iii) Stallings, W., *Operating Systems: Internals and Design Principles*, 9<sup>th</sup> edition, Pearson

Education, 2018.

(iv) Tanenbaum, A. S., *Modern Operating Systems*. 3<sup>rd</sup> edition. Pearson Education, 2007.

### **Suggested Practical List**

1. Usage of following commands: ls, pwd, cat, who, rm, mkdir, rmdir, cd.
2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, bc.
4. Write a shell script to display date in the mm/dd/yy format.
5. Write a shell script to display the multiplication table any number.
6. Write a shell script to find the factorial of a given number.
7. Program to show the pyramid of special character “\*”.
8. Write a shell script to find the sum of digits of a given number.
9. Write a shell script to perform the tasks of basic calculator.
10. Write a shell script to find the power of a given number.
11. Write a shell script to check whether the number is Armstrong or not.
12. Write a shell script to find the GCD (greatest common divisor) of two numbers.
13. Write a shell script to check if the number entered at the command line is prime or not.
14. Write a shell script to display on the screen sorted output of “who” command along with the total number of users.
15. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid”.
16. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
17. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
18. Write a shell script to check whether the file have all the permissions or not.
19. Write a shell script to modify “cal” command to display calendars of the specified months.
20. Write a shell script to modify “cal” command to display calendars of the specified range of months.

### **GE5b: Advanced Web Programming**

#### **Course Objective**

The course aims to familiarize the students with the concepts and techniques of server side web programming. This will enable the students to create dynamically generated web pages using HTML, PHP, MySQL and JQuery.

## Course Learning Outcomes

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

1. develop interactive and dynamic websites.
2. write programs to communicate with the server using GET and POST methods
3. learn to connect and manipulate databases using MySQL
4. validate server-side/backend data

## Syllabus

**Unit 1 Introduction to PHP:** Basic Syntax, defining variables and constants, data types, operators and expressions, decision making statements, constructs for iterations.

**Unit 2 String:** Handling: Creating a string and accessing its content, searching and replacing content of a string, and other built-in functions.

**Unit 3 Handling HTML Form with PHP:** Creating a form, submitting data to the server at the backend using GET and POST methods, GET vs POST methods.

**Unit 4 Database:** Connectivity with MySQL: Connectivity with database, database creation, creating tables, insertion and retrieval of the data from the database.

**Unit 5 jQuery and JSON:** Introduction to jQuery, syntax, selectors, events. JSON file format for storing and transporting data.

## References

1. Nixon, R. *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Reilly, 2018.
2. Holzner S. *PHP: The Complete Reference*, McGraw Hill, 2017
3. Murach J, *Murach's PHP and MySQL*, 2<sup>nd</sup> edition, Mike Murach & Associates, 2014.
4. Duckett, J. *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley, 2014.

## Suggested Practical List

1. Write a PHP script to reverse the digits of a number.
2. Create a web page containing two text boxes and a button named "Evaluate". When the user enters numbers in the text boxes and clicks on "Evaluate" button, a function should evaluate the sum of the numbers and display the result.
3. Write a script in PHP to display a Multiplication Table.

4. Write a script in PHP to display the following pattern:

```
1
22
333
4444
55555
```

Input number of rows from the user.

5. Write a PHP script to perform following string operations using in-built functions and built an interactive web page having buttons for each of the following operation:

- a. Find the length of a string
- b. Find a substring from a string
- c. Replace text within a string
- d. Remove whitespace and other predefined characters from both sides of a string.
- e. Check if a value is a string
- f. Convert the first character of each word in a string into uppercase.

6. Design a Login form and validate that form using PHP code. Display error message box when data is not valid otherwise redirect to the next page and display “Welcome username!”.

7. Design a student registration form, using appropriate input fields consisting of following:

- a. First Name
- b. Last Name
- c. Gender
- d. Roll Number
- e. Phone Number
- f. Course

Submit and retrieve the form data using \$\_POST, \$\_GET variable.

8. Write PHP Code to make connection to MySQL database, create database and tables and perform insertion, deletion, and retrieval of the data from the database. Display the messages such as “The record is added in the database!” when data is inserted into the database, “A record is deleted from the database” when data is deleted from the database. Use appropriate button names such as Add Data, Delete Data, and Display Data.

### **jQuery and JSON**

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery



4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

## **GE5c: Java Based Web App Development**

### **Course Objective**

Under Preparation

### **Course Learning Outcomes**

Under Preparation

### **Syllabus**

**Unit 1 Review of Programming Language:** Programming Constructs, Data types, Operators, Concepts of Class, Interface, Inheritance, Exception Handling, Util package, Multithreading, event handling.

**Unit 2 Java Database Connections:** Database connectivity, Connection , statement, result set object, Metadata, Connection pooling, CRUD operations, Prepared and callable statements

**Unit 3 Introduction to servlets:** Concepts of Streams, events and listener, recap of HTML, CSS, XML, Servlet package and interface, life cycle of servlet, deployment descriptor, Filters, HHTTP and Generic servlet, request dispatcher, Request Response classes, Dynamic page designing using servlet.

**Unit 4 Introduction to JSP:** JSP Life cycle, tags in JSP, custom tags, Expression Language, Introduction to Struts Framework, Implicit objects, database access using JSP

### **References**

1. Herbert Schildt, *Java : The Complete Reference*, 12<sup>th</sup> edition, McGraw-Hill Education, 2021.
2. Hans Bergsten, *Java Server Pages*, 3<sup>rd</sup> edition, O'Reilly, 2003.
3. Jim Keogh, *The Complete Reference J2EE*, 1<sup>st</sup> edition, McGraw-Hill Education, 2017.

### **Suggested Practical List**

Under Preparation

## GE6a: Computer Networks

### Course Objective

The course objectives of this paper are to: understand the concepts behind computer networks and data communication. learn the different types of networks, network topologies and their characteristics. learn the working of protocols used at various layers. understand the utility of different networking devices.

### Course Learning Outcomes

Upon successful completion of the course, students will be able to:

1. differentiate between various types of computer networks and their topologies.
2. understand the difference between the OSI and TCP/IP protocol suit.
3. distinguish between different types of network devices and their functions.
4. design/implement data link and network layer protocols in a simulated networking environment.

### Syllabus

**Unit 1 Introduction:** Types of computer networks, Internet, Intranet, network topologies (bus, star, ring, mesh, tree, hybrid topologies), network classifications. layered architecture approach, OSI Reference Model, TCP/IP Reference Model. Transmission Modes: simplex, half duplex and full duplex.

**Unit 2 Physical Layer:** Analog signal, digital signal, the maximum data rate of a channel, transmission media (guided transmission media, wireless transmission, satellite communication), multiplexing (frequency division multiplexing, time-division multiplexing, wavelength division multiplexing). Guided Media (Wired) (Twisted pair, Coaxial Cable, Fiber Optics). Unguided Media (Radio Waves, Infrared, Micro-wave, Satellite).

**Unit 3 Data Link and MAC Layer:** Data link layer services, error detection and correction techniques, error recovery protocols (stop and wait, go back n, selective repeat), multiple access protocols with collision detection, MAC addressing, Ethernet, data link layer switching, point-to-point protocol.

**Unit 4 Network layer:** Networks and Internetworks, virtual circuits and datagrams, addressing, subnetting, Dijkstra Routing algorithm, Distance vector routing, Introduction to Network Layer protocol- (ARP, IPV4, ICMP).

**Unit 5 Introduction to Transport and Application Layer:** Introduction to Process to process Delivery- (client-server paradigm, connectionless versus connection-oriented service); User Datagram Protocols, TCP/IP protocol, Flow Control. FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), Telnet (Remote login protocol), WWW (World Wide Web), HTTP (HyperText Transfer Protocol), URL (Uniform Resource Locator).

### References

1. Tanenbaum, A.S. & Wethrall, D.J., *Computer Networks*, 5<sup>th</sup> edition, Pearson Education, 2012.
2. Forouzan, B. A., *Data Communication and Networking*, 4<sup>th</sup> edition, McGraw-Hill Education, , 2017.

### Additional References

- (i) Comer, D. E., *Computer Networks and Internet*, 6<sup>th</sup> edition, Pearson Publication, 2015.
- (ii) Stallings, W., *Data and Computer Communications*, 10<sup>th</sup> edition, Pearson education India, 2017.

### Suggested Practical List

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm.
6. Simulate and implement the Dijkstra algorithm for shortest-path routing.

## **GE6b: Internet Technologies: Web App Design and Development**

### Course Objective

1. Develop understanding of Web Development Architecture.

2. Using React components in Web applications
3. Introduce REST APIs Design
4. Understanding of Angular Architecture, data-binding and dependency injection
5. Understand form validations and application of templates

### **Course Learning Outcomes**

On successful completion of the course students will be able to

1. Develop interfaces for single page applications
2. Develop a complete client side solutions using angular js
3. Develop a RESTful web services.
4. Apply form validations

### **Syllabus**

**Unit 1 Introduction to React:** Definition of React, React library, React Developer tools, Introduction to ES6, Declaring variables, Arrow Functions, Objects and Arrays, modules, Introduction to AJAX, Functions in AJAX Pure React: Page setup, virtual DOM, React Element, React DOM, Constructing Elements with Data, React Components, DOM Rendering, First React Application using Create React App, React with JSX, React Element as JSX Props, State and Component Tree: Property Validation, Validating Props with createClass, Default Props, ES6 Classes and stateless functional components, React state management, State within the component tree, state vs props, Forms in React

**Unit 2 Rest APIs:** JSON: Introduction, Syntax, Data Types, Objects, Schema. REST API: Introduction, WRML, REST API Design, Identifier Design with URIs, Interaction Design with HTTP, Representation Design, Caching, Security.

**Unit 3 Angular.js.:** Introduction to Angular: Angular architecture; introduction to components, component interaction and styles; templates, interpolation and directives; forms, user input, form validations; data binding and pipes; retrieving data using HTTP; Angular modules

### **References**

1. D. Brad, B. Dayley and C. Dayley, *Node.js, MongoDB and Angularjs Web Development: The definitive guide to using the MEAN stack to build web applications*, 2<sup>nd</sup> edition, Addison-Wesley, 2018.
2. D. Herron, *Node.js Web Development*, 5<sup>th</sup> edition, Packt Publishing, 2020.
3. A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, 1<sup>st</sup> edition, O'Reilly, 2017.
4. M. Masse, *REST API – Design Rulebook*, 1<sup>st</sup> edition, O'Reilly, 2011.

### **Suggested Practical List**

Under Preparation

## **GE6c: Artificial Intelligence**

### **Course Objective**

This course introduces the basic concepts and techniques of Artificial Intelligence (AI) along with the foundations of AI. Artificial Intelligence is the branch of computer science devoted to the study and development of methods that enable computers to do things that would require intelligence if done by humans. The focus of this course is to impart knowledge about the use of core AI techniques having applicability to a wide range of real-world problems. Topics covered comprise an introduction to AI and intelligent agents, concepts of learning and reasoning, search techniques, game playing, knowledge representation and understanding of natural languages.

### **Course Learning Outcomes**

On successful completion of this course, students will be able to:

1. identify problems that are amenable to solutions by specific AI methods.
2. knowledge of basic Artificial Intelligence algorithms, such as search problems, minimax algorithm etc.
3. demonstrate working knowledge in Prolog by writing simple Prolog programs.
4. appreciate the utility of different types of AI agents.

### **Syllabus**

**Unit 1 Introduction:** Introduction to artificial intelligence, background and applications, Turing test, Software agents, Rational agents, Intelligent agents, structure of various agents, behavior and environment for agents.

**Unit 2 Problem Solving and Searching Techniques:** Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A\* algorithm, constraint

satisfaction problem, Introduction to game playing, min-max and alpha-beta pruning algorithms.

**Unit 3 Knowledge Representation:** Propositional logic, First-Order Predicate Logic, resolution principle, unification, Associate Networks, conceptual dependencies, frames, and scripts, Introduction to Programming in Logic (PROLOG).

**Unit 4 Understanding Natural Languages:** Components and steps of communication, contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free Grammars.

**Unit 5 AI- The Present and Future:** Weak AI, Strong AI, Human-level AI Models, Data-driven Models and Machine Learning End-to-end Deep Learning AI Models.

### Text Books

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4<sup>th</sup> edition, 2020.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3<sup>rd</sup> edition, Tata McGraw Hill, 2010.
3. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4<sup>th</sup> edition, 2012.

### Reference Books

- (i) Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
- (ii) Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
- (iii) Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5<sup>th</sup> edition, Springer, 2003.

### Suggested Practical List

1. Write a program to implement two agents to communicate with each other with message passing in Python/Prolog.
2. Write a program to implement the Hill climbing search algorithm in Python/Prolog.
3. Write a program to implement the Best first search algorithm in Python/Prolog.
4. Write a program to implement A\* search algorithm in Python/Prolog.
5. Write a program to implement the min-max search algorithm in Python/Prolog.
6. Write a program to solve the Water-Jug Problem in Python/Prolog.

7. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Python/Prolog.
8. Write a prolog program to implement the family tree and demonstrate the family relationship.
9. Write a prolog program to implement knowledge representation using frames with appropriate examples.
10. Write a Prolog program to implement `conc(L1, L2, L3)` where L2 is the list to be appended with L1 to get the resulted list L3.
11. Write a Prolog program to implement `reverse(L, R)` where List L is original and List R is reversed list.
12. Write a PROLOG program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
13. Write a prolog program to recognize context free grammar  $a^n b^n$ .

### **GE7a: Information Security**

#### **Course Objective**

The goal of this course is to make a student learn basic principles of information security. Over the due course of time, the student will be familiarized with cryptography, authentication and access control methods along with software security. Potential security threats and vulnerabilities of systems are also discussed along with their impacts and countermeasures. This course also touches upon the implications of security in cloud and Internet of Things (IoT).

#### **Course Learning Outcomes**

On successful completion of this course, a student will be able to

1. Identify the major types of threats to information security.
2. Describe the role of cryptography in security.
3. Discover the strengths and weaknesses of private and public key cryptosystems.
4. Identify and apply various access control and authentication mechanisms.
5. Discuss data and software security and related issues.
6. Explain network security threats and attacks.
7. Articulate the need for security in cloud and IoT.

**Unit 1 Overview:** Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles, Attack Surfaces and Attack Trees.

**Unit 2 Cryptographic tools:** Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers, Practical Application: Encryption of Stored Data.

**Unit 3 Data Security:** User authentication and Access Control, Database and Data Center Security

**Unit 4 Software Security:** Types of Malicious Software, Threats, Viruses, Worms, SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing, Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. Overflow Attacks - Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks. Handling Program Input, Writing Safe Program Code, Interacting with the Operating System and Other Programs.

**Unit 5 Network Security:** Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, Firewalls, Secure Email and S/MIME, Secure Sockets Layer (SSL) and Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security, Public-Key Infrastructure.

**Unit 6 Wireless, Cloud and IoT Security:** Cloud Computing, Cloud Security Concepts, Cloud Security Approaches, The Internet of Things, IoT Security. Wireless Security Overview, Mobile Device Security, IEEE 802.11 Wireless LAN.

## References

1. Stallings, W. and Brown L., *Computer Security: Principles and Practice*, 4<sup>th</sup> edition, Pearson education, 2018.

## Additional References

1. Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5th edition, Prentice Hall, 2015.
2. Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications*, 2nd edition, Pearson Education, 2004.
3. Stallings W. *Cryptography and network security*, 7th edition, Pearson Education, 2018.
4. Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.



5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised Edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd Edition, John Wiley & Sons, 2008.

### **Suggested Practical List**

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyse a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

## **GE7b: Design and Analysis of Algorithms**

### **Course Objective**

The course is designed to develop understanding of different algorithm design techniques and use them for problem solving. The course shall also enable the students to verify correctness of algorithms and analyze their time complexity.

### **Course Learning Outcomes**

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

1. Compute and compare the asymptotic time complexity of algorithms.
2. Use appropriate algorithm design technique(s) for solving a given problem.
3. Apply hashing and collision resolution techniques.

### **Syllabus**

**Unit 1 Sorting, Selection:** Insertion Sort, Linear Time Sorting - Count Sort, Radix Sort, Selection Problem. Analysis of time complexity of all algorithms.

**Unit 2 Graphs:** Graph, representation of graphs, traversal of graphs, directed graphs, Directed Acyclic Graphs and Topological Ordering; all with analysis of time complexity.

**Unit 3 Divide and Conquer:** Introduction to divide and conquer technique, Merge Sort, Quick Sort with analysis of time complexity.

**Unit 4 Greedy algorithms:** Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, shortest path problem with analysis of time complexity.

**Unit 5 Dynamic Programming:** Introduction to the Dynamic Programming approach, application to weighted interval scheduling, integer knapsack problem with analysis of time complexity.

**Unit 6 Hash Tables:** Hash Functions, Collision resolution schemes.

## References

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C., *Introduction to Algorithms*, 4<sup>th</sup> edition. Prentice Hall of India. 2022.
2. Kleinberg, J., Tardos, E., *Algorithm Design*, 1<sup>st</sup> edition, Pearson, 2013.

## Additional References

- (i) Basse, S., Gelder, A. V. *Computer Algorithms: Introduction to Design and Analysis*. 3<sup>rd</sup> edition, Pearson, 1999.

## Suggested Practical List

1. Implement Insertion Sort, Selection sort, Bubble Sort (The program should report the number of comparisons).
2. Implement Merge Sort (The program should report the number of comparisons).
3. Implement Quick sort (The program should report the number of comparisons).
4. Implement Radix Sort.
5. Implement Count Sort.
6. Implement Searching Techniques: Linear and binary.
7. Implement BFS traversal on a graph.
8. Implement DFS traversal on a graph.
9. Implement 0-1 knapsack problem using DP

## GE7c: Internet Technologies - Mobile App Design and Development

### Course Objective

Under Preparation

### Course Learning Outcomes

Under Preparation

### Syllabus

**Unit 1 Android Systems:** Introduction to Mobile devices and applications, Open Handset Alliance (OHA), Overview of Android OS and architecture, installing android studio. Introduction to Android application components, Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions Activities and intents: understanding activity and its life cycle, Types of intents, intent filter, context, data sharing using intent

**Unit 2 Android User Interface:** Basic android UI, layouts, view and view attributes, button, controls. UI events and event listeners, animations, notifications, progress dialog, Action bar, toolbar, menus and pop ups, Tab based UI, Fragment, Types of Fragment, Fragment Lifecycle, communication between fragment and activity

**Unit 3 Android Storage and APIs:** Android storage: Using Android Data and Storage APIs, Managing data using SQLite, Sharing Data between Applications with Content Providers  
Android APIs: Multimedia, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, android location based services

**Unit 4 iOS Technology Stack:** Introduction to iOS technology stack: iOS architecture, StoryBoard, features of Xcode, components of iOS SDK. Introduction to swift: data types, variables, control flow and operators, Collections and functions in swift, classes and structures, inheritance, closure and enumerations

**Unit 5 User interactions:** Controls, gesture organizers, touching views, Core Location and Mapkit, using Google Maps in iOS. Sensors in iOS. Data persistence: Core Data framework for storing persistent data, CRUD operations.

Note: Kotlin will be used for the implementation.

### References

1. Meier Reto and Ian Lake, *Professional Android*, 4<sup>th</sup> edition, Wrox, 2018.

2. Craig Grummitt, *iOS Development with Swift*, Manning publications.
3. Rick Boyer, *Android 9 Development Cookbook*, Packt Publishing Limited, 2018.

### **Suggested Practical List**

Under Preparation

## **GE 8a: Machine Learning**

### **Course Prerequisites**

A course in probability, statistics, linear algebra and multivariate analysis

### **Course Objective**

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

### **Course Learning Outcomes**

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

1. Differentiate between supervised and unsupervised learning tasks.
2. Appreciate the need of preprocessing, feature scaling and feature selection.
3. Understand the fundamentals of classification, regression and clustering
4. Implement various machine learning algorithms learnt in the course.

### **Syllabus**

**Unit 1 Introduction:** Basic definitions and concepts, key elements, supervised and unsupervised learning, introduction to reinforcement learning, applications of ML.

**Unit 2 Preprocessing:** Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis).

**Unit 3 Regression:** Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

**Unit 4 Classification:** Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics.

**Unit 5 Clustering:** Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

## References

1. Mitchell, T.M. *Machine Learning*, McGraw Hill Education, 2017.
2. James, G., Witten. D., Hastie. T., Tibshirani., R. *An Introduction to Statistical Learning with Applications in R*, Springer, 2014.
3. Alpaydin, E. *Introduction to Machine Learning*, MIT press, 2009.

## Additional References

- (i) Flach, P. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, 2015.
- (ii) Christopher & Bishop, M. *Pattern Recognition and Machine Learning*, New York: Springer-Verlag, 2016.
- (iii) Sebastian Raschka, *Python Machine Learning*, Packt Publishing Ltd., 2019.

## Suggested Practical List

Use Python for practical labs for Machine Learning. Utilize publically available datasets from online repositories like <https://data.gov.in/> and <https://archive.ics.uci.edu/ml/datasets.php>

For evaluation of the regression/classification models, perform experiments as follows:

- Scale/Normalize the data
- Reduce dimension of the data with different feature selection techniques
- Split datasets into training and test sets and evaluate the decision models
- Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows:

- MSE and  $R^2$  score for regression models
- Accuracy, TP, TN, FP, FN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier
2. Simple Linear Regression multiple linear regression
3. Polynomial Regression
4. Lasso and Ridge Regression
5. Logistic regression
6. Artificial Neural Network
7.  $k$ -NN classifier
8. Decision tree classification
9. SVM classification
10. K-Means Clustering
11. Hierarchical Clustering

### **GE8b: Digital Marketing and Social Media Analytics**

Under Preparation

### **GE8c: Introduction to Parallel Programming**

#### **Course Prerequisites**

Programming in C++, Data Structures, Computer System Architecture, Operating Systems

#### **Course Objective**

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

#### **Course Learning Outcomes**

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

1. Appreciate the need of Parallel algorithms
2. Describe architectures for parallel and distributed systems.
3. Develop elementary parallel algorithms in shared memory models.
4. Develop elementary parallel algorithms in distributed memory models.

## Syllabus

**Unit 1 Introduction to Parallel Computing:** Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

**Unit 2 OpenMP programming for shared memory systems:** Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model, Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks.

**Unit 3 MPI programming for distributed memory systems:** MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, Introduction to MPI Collectives, Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

**Unit 4 Applications:** Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm.

## References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2<sup>nd</sup> edition, Addison-Wesley, 2003.
2. Quinn, M., *Parallel Programming in C with MPI and OpenMP*, 1<sup>st</sup> Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016

## Suggested Practical List

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP
2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP

4. Implement Dijkstra's Algorithm in parallel using OpenMP

DRAFT



**Department of Computer Science  
University of Delhi**

**Skill Enhancement Courses (SECs)**

**Under UGCF 2022**

**DRAFT**

**Approved in  
UG Committee meeting held on May17, 2022  
Faculty of Mathematical Sciences meeting held on May 25, 2022**

1. [List of SECs](#)

**Table of SECs**

Semester	SEC -No.	Title	L	T	P	Total credits	Pre-requisites
I/III/V/V II	<a href="#">SEC01A</a>	Front End Web Design and Development	1	0	1	2	Nil
	<a href="#">SEC01B</a>	Office Automation Tools	1	0	1	2	Nil
	<a href="#">SEC01C</a>	App Development for Android	1	0	1	2	Basic knowledge of programming
	<a href="#">SEC1D</a>	Basics of Programming in Python	1	0	1	2	For non-CS students
II/IV/VI/ VIII	<a href="#">SEC02A</a>	Backend Web Development	1	0	1	2	Knowledge of HTML, CSS
	<a href="#">SEC02B</a>	Data Analysis using Spreadsheet	1	0	1	2	Nil
	<a href="#">SEC02C</a>	App Development for iOS	1	0	1	2	Basic knowledge of programming
	<a href="#">SEC2D</a>	Introduction to Object Oriented Programming using Python	1	0	1	2	Basic knowledge of Python

**Note:**

1. For a prerequisite of Python/HTML with CSS, a course at plus 2 level is acceptable.
2. Batch size for Practicals will be (8-10) and Tutorials will be (12-15).
3. More SECs may be added in due course of time.

**SEC01A: Front End Web Design and Development****Course Objective**

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple, interactive and stylish websites using HTML, CSS and Javascript.

**Course Learning Outcomes**

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

1. Build websites using the elements of HTML.
2. Build interactive and stylish websites using the client side programming techniques with CSS and Javascript.
3. Learn to validate client-side data.
4. Define the structure and contents of the website using different features of CSS.

**Syllabus**

**Unit 1 Introduction:** Introduction to internet and web design. Basic concepts of web architecture.

**Unit 2 HTML:** Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames.

**Unit 3 Cascading style sheet (CSS):** Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding properties,

margin properties).

**Unit 4 Basics of Javascript:** Document object model, data types and variables, functions, methods and events, controlling program flow, built-in objects and operators, validations.

## References

1. Nixon, R., *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Reilly, 2018.
2. Powell, T.A. *HTML & CSS: The Complete Reference*, 5<sup>th</sup> edition, Tata McGrawHill, 2017.
3. Duckett, J., *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley, 2014.

## Additional References

- (i) Boehm, A., & Ruvalcaba, Z., *Murach's HTML5 and CCS*, 4<sup>th</sup> edition, Mike Murach & Associates, 2018.
- (ii) Ivan Bayross, *Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI*, BPB Publications, 2010.

## Suggested Practical List

### HTML

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create an HTML document for displaying the current semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame needs to show your college's name and logo. Bottom horizontal frame is to split into two vertical frames. The left frame has hyperlinks to pages related to faculty, courses, student activities, etc. The right frame shows the corresponding webpage based on the link

clicked on the left frame.

5. Create a student registration form using HTML which has the following controls and make interactive content presentation using CSS.:
  - I. Text Box
  - II. Dropdown box
  - III. Option/radio buttons
  - IV. Check boxes
  - V. Reset and Submit button
6. Create a webpage for your department with a drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors etc. learnt in CSS
7. Write event driven programs in JavaScript for the following:
  1. Enter a number and on click of a button print its multiplication table.
  2. Print the largest of three numbers entered by the user.
  3. Find the factorial of a number entered by the user.
  4. Enter a list of positive numbers using the prompt terminated by a zero. Find the sum and average of these numbers.
8. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customize the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:
  - a. Roll number is a 7-digit numeric value
  - b. Name should be an alphabetical value(String)
  - c. Non-empty and valid fields like DOB

## **SEC01B: Office Automation Tools**

### **Course Objective**

The course introduces the students to document processing, presentation software and data handling. The basic features and skills of creating, editing, inserting tables, graphics as well as presentation tools along with spreadsheet data handling are covered.

## Course Learning Outcomes

On successful completion of this course, a student will be able to:

1. create and refine documents using text formatting, tables and graphics.
2. use mail merge.
3. create macros and templates in documents.
4. protect documents.
5. create presentations containing transitions and animations. learn advanced presentation features like custom slide show, call outs and action buttons.
6. use referencing and functions for data handling.

**Unit 1 Document Processing Basics:** Creating, opening and saving a document, text formatting, header and footer, creating and editing of tables, importing graphics, insert picture, using word processor's drawing features, text in drawing.

**Unit 2 Advanced Features:** Creating macros, watermarks, templates, reviewing documents, comparing and combining documents, protection of documents-using passwords. Mail merge concept, main document, data sources, merging data source and main document.

**Unit 3 Presentation Tools:** Creating presentations, using blank presentation option, using design template option, adding slides, deleting a slide, importing images from the outside world, deleting a slide, numbering a slide, saving presentation transition and animations, adding notes to slides, customize slideshow.

**Unit 4 Data Handling using spreadsheets:** Relative, absolute and mixed referencing, mathematical and statistical functions, nested functions

## References

1. Mali, L., *Libre office 5.1 Impress, Draw*. Base book- Vol 2, Notion Press, 2017..
2. Libreoffice Documentation Team, *Getting Started with LibreOffice 6.0*. Friends of OpenDocument, Inc., 2018.
3. Libreoffice Documentation Team, *LibreOffice 6.0 Writer Guide*. Friends of OpenDocument, Inc., 2018.

## Suggested Practical List

To be done according to the above topics.

## SEC01C: App Development for Android

### Course Objective

This course is designed as the first course for developing simple mobile applications that can run on Android mobiles and tablets. The students will learn the fundamentals of Android OS. They will be able to write applications which can work with databases and store the data locally.

### Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Install and configure Android app development tools.
2. Design user interfaces for android apps.
3. Handle mobile storage through apps.
4. Design apps to handle SQLite database

### Syllabus

**Unit 1 Android Systems:** Introduction to Mobile devices and applications, Open Handset Alliance (OHA), Overview of Android OS and architecture, installing android studio. Introduction to Android application components, Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions Activities and intents: understanding activity and its life cycle, Types of intents, intent filter, context, data sharing using intent

**Unit 2 Android User Interface:** Basic android UI, layouts, view and view attributes, button, controls. UI events and event listeners, animations, notifications, progress dialog, Action bar, toolbar, menus and pop ups, Tab based UI, Fragment, Types of Fragment, Fragment Lifecycle, communication between fragment and activity

**Unit 3 Android Storage and APIs:** Android storage: Using Android Data and Storage APIs, Managing data using SQLite, Sharing Data between Applications with Content Providers  
Android APIs: Multimedia, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, android location based services.

Note: Kotlin will be used for the implementation.

## References

1. Meier Reto and Ian Lake, *Professional Android*, 4<sup>th</sup> edition, Wrox, 2018.
2. Rick Boyer, *Android 9 Development Cookbook*, Packt Publishing Limited, 2018.

## SEC01D: Basics of Programming in Python

### Course Objective

This course is designed as the first course that introduces programming concepts using Python. The course focuses on the development of Python programming to simple solve problems of different domains.

### Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Understand the basics of programming language
2. Develop, document, and debug modular Python programs.
3. Apply suitable programming constructs and built-in data structures to solve a problem.

### Syllabus

**Unit 1 Introduction to Programming:** Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

**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.

**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.

## References



1. Taneja, S., Kumar, N. *Python Programming- A modular Approach*, Pearson Education India, 2018.
2. Balaguruswamy E., *Introduction to Computing and Problem Solving using Python*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.

### **Additional References**

1. Brown, Martin C., *Python: The Complete Reference*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.
2. Guttag, J.V., *Introduction to computation and programming using Python*, 2<sup>nd</sup> edition, MIT Press, 2016.

### **Suggested Practical List**

1. WAP to find the roots of a quadratic equation
2. WAP to accept a number 'n' and
  - a. Check if 'n' is prime
  - b. Generate all prime numbers till 'n'
  - c. Generate first 'n' prime numbers
  - d. This program may be done using functions
3. WAP to create a pyramid of the character '\*' and a reverse pyramid

```
*
***
*****
*****
*****
*****
*****
*****
***
*
```

4. WAP that accepts a character and performs the following:
  - a. print whether the character is a letter or numeric digit or a special character
  - b. if the character is a letter, print whether the letter is uppercase or lowercase
  - c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. WAP to perform the following operations on a string

- a. Find the frequency of a character in a string.
  - b. Replace a character by another character in a string.
  - c. Remove the first occurrence of a character from a string.
  - d. Remove all occurrences of a character from a string.
6. WAP to swap the first n characters of two strings.
  7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.
  8. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:
    - a. 'for' loop
    - b. list comprehension

## SEC02A: Backend Web Development

### Course Objective

The course aims at introducing the basic concepts and techniques of server side web programming. The student shall be able to develop interactive and dynamic websites using HTML, PHP and MySQL.

### Course Learning Outcomes

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

1. build interactive and dynamic websites.
2. write the server side programming techniques with PHP for accessing the contents to/from the server
3. learn to validate server-side/backend data
4. use GET and POST methods for sending data within client and server.

**Unit 1 Introduction to PHP:** Basic Syntax, defining variables and constants, data types,

operators and expression, decision making statements, loop making constructs, mixing decisions and looping with HTML

**Unit 2 String Handling:** Creating a string and accessing its content, searching and replacing content of a string, formatting a string, string built-in-functions

**Unit 3 Handling HTML Form with PHP:** Creating a form, submitting data to the server at the backend using GET and POST methods, GET vs POST methods, create a student' registration form

**Unit 4 Database Connectivity with MySQL:** Connectivity with database, database creation, creating tables, insertion and retrieval of the data from the database, data manipulation.

## References

1. Nixon, R., *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Reilly, 2018.
2. Murach J, Murach's, *PHP and MySQL*, 2nd Edition, Mike Murach & Associates, 2014.
3. Holzner S., *PHP: The Complete Reference*, McGraw Hill, 2017.

## Suggested Practical List

1. Write a PHP script to print the sum of odd digits of a number.
2. Create a web page having two radio buttons. One for checking whether the given year is a Leap Year or not and another button to check whether the given number is a Palindrome or not. Write a PHP script for the functionality of each button.
3. Write a script in PHP to display a Multiplication Table using nested for loop.
4. Write a script in PHP to display simple Pyramid pattern:

```
*
* *
* * *
* * * *
* * * * *
```
5. Write a PHP script to perform following string operations using in-built functions and built an interactive web page having buttons for each of the following operation:
  - a. Find the length of a string
  - b. Find a substring from a string
  - c. Replace text within a string
  - d. Remove whitespace and other predefined characters from both sides of a string.

- e. Check if a value is a string
  - f. Reverse a given string
  - g. Convert the first character of each word in a string into uppercase.
6. Design a Login form and validate that form using PHP code (Server side validation). Display message box when data is empty otherwise redirect to next page and display “Welcome username!”.
  7. Design a student registration form, using appropriate input fields consisting of following:
    - a. First Name
    - b. Last Name
    - c. Gender
    - d. Roll Number
    - e. Department
    - f. SocietiesSubmit and retrieve the form data using \$\_POST, \$\_GET variable.
  8. Write PHP Code to make connection to MySql database, create database and tables and access their contents.
  9. Write PHP code to insert, delete, and select the data from the database. Create proper forms for performing the above operations. Display the messages such as “The record is added in the database!” when data is inserted into the database, “A record is deleted from the database” when data is deleted from the database. Use appropriate button names such as Add Data, Delete Data, and Display Data.

## **SEC02B: Data Analysis using Spreadsheet**

### **Course Objective**

The course will help students to learn how to analyze data with spreadsheets. They will learn about referencing, charts, functions and various utilities.

### **Course Learning Outcomes**

On successful completion of this course, a student will be able to:

1. perform data analysis and manipulation in a spreadsheet.
2. use built-in mathematical functions in a spreadsheet.
3. perform what-if analysis using Goal seek, ASAP utility add-ins in spreadsheets.
4. sort and filter data.
5. protect a spreadsheet

**Unit 1 Introduction to Spreadsheets:** workbook and worksheets; relative, absolute and mixed referencing; mathematical, statistical and database functions, nested functions, regular expressions in functions; pivot table.

**Unit 2 Charts:** Data visualization using built-in charts.

**Unit 3 Utilities:** What-if scenarios, goal-seek, solver, data validation, creating a dropdown list from a range of cells, data filtering and sorting, calculations using linked sheets, detective tools.

**Unit 4 Protection:** passwords and digital signatures in Spreadsheets.

## References

1. Mali, L., *Libre Office 5.1 Writer, Calc: Math Formula Book* - Vol 1. Notion Press, 2017.
2. Libre Office 7.0 Calc Guide.
3. Libre Office 4.1 Calc Guide The Document Foundation.

## Suggested Practical List

To be done according to the above topics.

## SEC02C: App Development for iOS

### Course Objective

This course is designed to enable a student to learn basics about iOS app development for iPhones/ iPads/ iPods. The course provides an overview of Swift programming language, which is used to design iOS apps. The students will be able to design apps with simple user interface. They will also be able to handle sensors like camera, microphones and GPS through these apps.

### Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Learn about the Xcode app architecture

2. Define key programming terms relevant to Swift programming language
3. Develop iOS apps for user interfaces
4. Explain iOS API features to including location, sensors and gestures

## Syllabus

**Unit 1 iOS Technology Stack:** Introduction to iOS technology stack: iOS architecture, StoryBoard, features of Xcode, components of iOS SDK. Introduction to swift: data types, variables, control flow and operators, Collections and functions in swift, classes and structures, inheritance, closure and. enumerations

**Unit 2 User interactions:** Controls, gesture organizers, touching views, Core Location and Mapkit, using Google Maps in iOS. Sensors in iOS. Data persistence: Core Data framework for storing persistent data, CRUD operations.

## References

1. Craig Grummitt, *iOS Development with Swift*, Manning publications.

## SEC02D: Introduction to Object Oriented Programming using Python

### Course Objective

This course is designed for advanced programming concepts using Python. The course focuses on the development of Python programming using built-in data structures like lists, sets, tuples and dictionaries to solve problems of different domains. It also introduces the concept of object-oriented programming and file handling.

### Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Develop, document, and debug modular Python programs.
2. Apply suitable programming constructs and built-in data structures to solve a problem.
3. Use and apply various data objects in Python.

4. Use classes and objects in application programs and handle data files.

## Syllabus

**Unit 1 Built-in data structures:** List operations, tuples, sets, dictionaries and their operations, passing data structures to a user-defined function.

**Unit 2 Object Oriented Programming:** Introduction to classes, objects and methods; Standard libraries.

**Unit 3 File and exception handling:** File handling through libraries; Errors and exception handling.

## References

1. Taneja, S., Kumar, N. *Python Programming- A modular Approach*. Pearson Education India, 2018
2. Balaguruswamy E. *Introduction to Computing and Problem Solving using Python*, 2nd edition, McGraw Hill Education, 2018

## Additional References

1. Brown, Martin C. *Python: The Complete Reference*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.
2. Guttag, J.V. *Introduction to computation and programming using Python*, 2<sup>nd</sup> edition. MIT Press, 2016.

## Suggested Practical List

1. Write a function that accepts two lists and returns the merged list in sorted order
2. WAP to read a file and
  - a. Print the total number of characters, words and lines in the file.
  - b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
  - c. Print the words in reverse order.

- d. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.
3. WAP to define a class Point with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method distance to calculate the distance between any two point objects.
4. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
5. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:
  - a. Print half the values of the tuple in one line and the other half in the next line.
  - b. Print another tuple whose values are even numbers in the given tuple.
  - c. Concatenate a tuple t2=(11,13,15) with t1.
  - d. Return maximum and minimum value from this tuple
6. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.





# Digital Empowerment

VAC- Value Addition Course  
(Undergraduate Courses)  
University of Delhi

# Credit distribution, Eligibility and Pre-requisites of the Course

Course Title and Code	Credits	Credit Distribution of the Course			Eligibility Criteria	Prerequisite of the Course
<b>Digital Empowerment</b>	<b>02</b>	Lecture	Tutorial	Practical/ Practice	Pass in Class 12th	NIL
		<b>1</b>	<b>0</b>	<b>1</b>		

# Learning Objectives

The Learning Objectives of this course are as follows:

- Understand the digital world and need for digital empowerment
- Create awareness about Digital India.
- Explore, communicate and collaborate in cyberspace.
- Imparting awareness on cyber safety and security.

# Learning Outcomes

The Learning Outcomes of this course are as follows:

- Use ICT and digital services in daily life.
- Develop skills to communicate and collaborate in cyberspace using social platforms, teaching/learning tools.
- Understand the significance of security and privacy in the digital world.
- Recognizing ethical issues in the cyber world.

# Digital Empowerment Syllabus

Unit	Unit name	Topic
I	Digital inclusion and Digital Empowerment	Needs and challenges, Vision of Digital India
		DigiLocker, E-Hospitals, e-Pathshala, BHIM, e-Kranti (Electronic Delivery of Services), e-Health Campaigns
		Public utility portals of Govt. of India such as RTI, Health, Finance, Income Tax filing, Education
II	Communication and Collaboration in the Cyberspace	Electronic Communication: electronic mail, blogs, social media
		Collaborative Digital platforms
		Tools/platforms for online learning, Collaboration using file sharing, messaging, video conferencing
III	Towards Safe and Secure Cyberspace	Online security and privacy
		Threats in the digital world: Data breach and Cyber Attacks
		Blockchain Technology
		Security Initiatives by the Govt of India
IV	Ethical Issues in Digital World	Netiquettes
		Ethics in digital communication, Ethics in Cyberspace

# Practicals / Activities Component

- The course should be conducted in an interactive mode through demonstration, using appropriate tools.
- Conduct workshops on-services initiated under Digital India.
- Spread digital literacy/awareness amongst the vulnerable groups and marginalized sections of the society like street vendors, domestic help, security guards, senior citizens.
- Students will take up team activities/ projects exploring digital services in the areas such as education, health, planning, farming, security, cyber security, financial inclusion, and justice, e-Kranti.
- Any other Practical/Practice as decided from time to time.

# Operational Guidelines

- At least 5 Value Addition Courses from the pool of VACs may be offered in the College
- The Colleges shall make teaching arrangements for VACs by involving teachers belonging to the relevant/related discipline and/or having expertise/prior experience relevant to the course. Visiting/Special lectures, wherever required, may be arranged by the colleges for teaching the Courses.

# Organizational Structure

## At University Level

### VAC COMMITTEE

- Body for making policy decisions on VACs

### STEERING COMMITTEE

- For operation, implementation and monitoring of courses

### COURSE COMMITTEES

- To ensure quality and uniformity in pedagogy

## At College Level

### NODAL OFFICER

- Will coordinate with University Level Committees

### VAC COMMITTEE

- Three Members Committee
- Will oversee Implementation of VACs in the college.



# Unit 1: Digital inclusion and Digital Empowerment

Unit	Unit name	Week No	Topic	Teaching Resources
I	Digital inclusion and Digital Empowerment	1	Needs and challenges, Vision of Digital India	<a href="https://www.digitalindia.gov.in">https://www.digitalindia.gov.in</a>
		2-3	DigiLocker, E-Hospitals, e-Pathshala, BHIM, e-Kranti (Electronic Delivery of Services), e-Health Campaigns	<a href="https://digilocker.gov.in/">https://digilocker.gov.in/</a> <a href="https://nha.gov.in/">https://nha.gov.in/</a> <a href="https://pmjay.gov.in/">https://pmjay.gov.in/</a> <a href="https://diksha.gov.in/">https://diksha.gov.in/</a> <a href="https://swayam.gov.in/">https://swayam.gov.in/</a> <a href="https://digitalindia.gov.in/content/ekranti">https://digitalindia.gov.in/content/ekranti</a>
		4-5	Public utility portals of Govt. of India such as RTI, Health, Finance, Income Tax filing, Education	<a href="https://www.india.gov.in/">https://www.india.gov.in/</a> <a href="https://rtionline.gov.in/index.php">https://rtionline.gov.in/index.php</a> <a href="https://www.bhimupi.org.in/">https://www.bhimupi.org.in/</a> <a href="https://www.incometax.gov.in/iec/foportal">https://www.incometax.gov.in/iec/foportal</a> <a href="https://nha.gov.in/">https://nha.gov.in/</a> <a href="https://swayam.gov.in/">https://swayam.gov.in/</a> <a href="https://nptel.ac.in/">https://nptel.ac.in/</a>

# Unit 1: Digital Inclusion and Digital Empowerment

(Cont.)

Introduction to basic terminology:

- Digital world
- Digital citizen
- Digital inclusion
- Digital divide
- Digital empowerment

## Need

- Sustainable economy
- Healthcare
- Educational Achievements
- Efficient Civic Engagements

## Challenges

- Digital literacy among vulnerable groups
- ICT infrastructure
- Data Security
- e-Waste management

# Digital India (<https://www.digitalindia.gov.in>)

Vision: *“The vision of Digital India programme is to transform India into a digitally empowered society and knowledge economy.”*

The screenshot displays the Digital India website's 'Vision and Vision Areas' page. The browser address bar shows the URL [digitalindia.gov.in/content/vision-and-vision-areas#collapseone--13](https://www.digitalindia.gov.in/content/vision-and-vision-areas#collapseone--13). The page features a blue header with the Digital India logo and navigation menu items: HOME, ABOUT DI, DI INITIATIVES, ECOSYSTEM, NEWSTRACK, OPPORTUNITIES, and FEEDBACK. The main content area has a dark background with the title 'Vision and Vision Areas' and several icons representing digital connectivity. A sidebar on the left contains a table of contents with 'Vision and Vision Areas' highlighted. The main text area includes the heading 'Vision of Digital India' followed by the vision statement: 'The vision of Digital India programme is to transform India into a digitally empowered society and knowledge economy.' Below this is the heading 'Vision Areas of Digital India' and the text 'The Digital India programme is centred on three key vision areas:', which is followed by three partially visible content boxes.

digitalindia.gov.in/content/vision-and-vision-areas#collapseone--13

Contrast: High | Normal

Skip to main content

HOME ABOUT DI DI INITIATIVES ECOSYSTEM NEWSTRACK OPPORTUNITIES FEEDBACK

**Vision and Vision Areas**

Home → Vision and Vision Areas

**Vision of Digital India**

The vision of Digital India programme is to transform India into a digitally empowered society and knowledge economy.

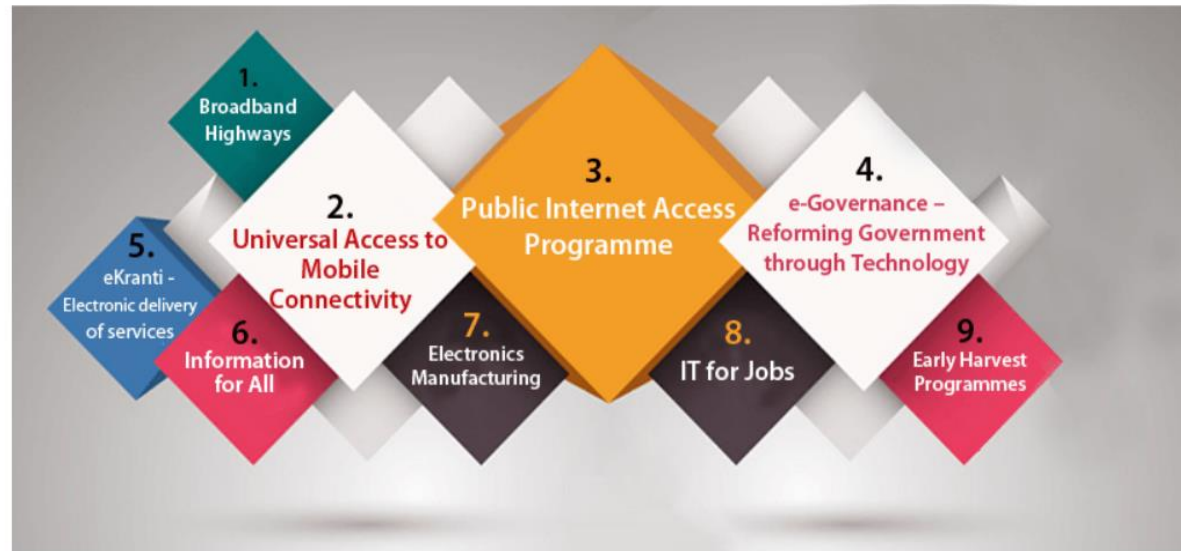
**Vision Areas of Digital India**

The Digital India programme is centred on three key vision areas:

# Digital India (<https://www.digitalindia.gov.in>)



## How Digital India will be realized: Pillars of Digital India



Digital India is an umbrella programme that covers multiple Government Ministries and Departments. It weaves together a large number of ideas and thoughts into a single, comprehensive vision so that each of them can be implemented as part of a larger goal.

Each individual element stands on its own, but is also part of the larger picture. Digital India is to be implemented by the entire Government with overall coordination being done by the Department of Electronics and Information Technology (DeitY).

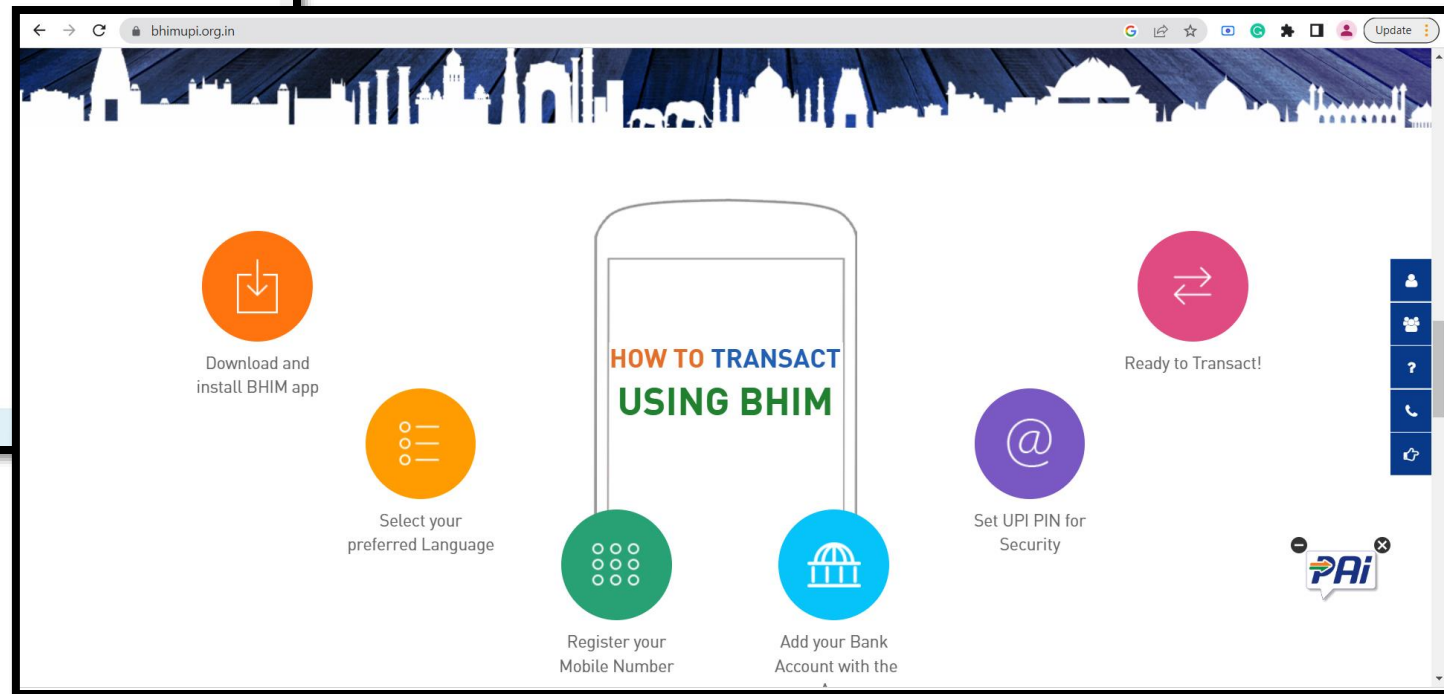
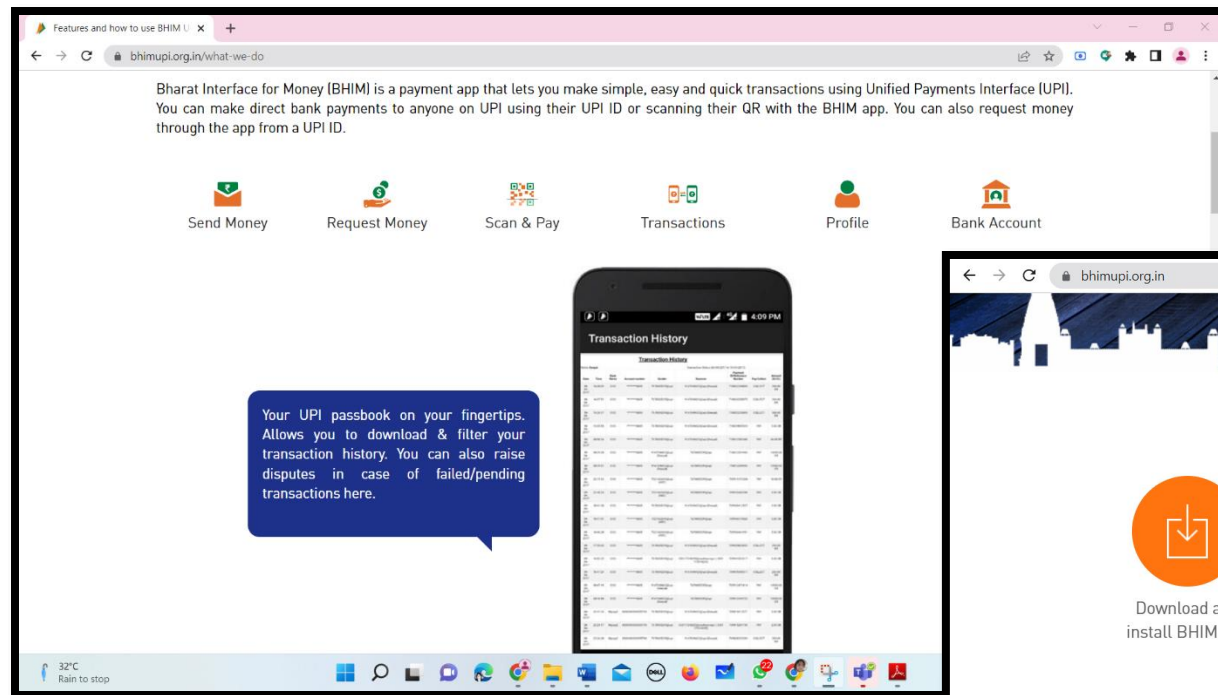


# DigiLocker (<https://www.digilocker.gov.in/>)

The image shows two screenshots of the DigiLocker website. The top screenshot displays the homepage with a banner for 'CBSE 2022 Class XII Marksheet' and a 'Get Yours' button. Below the banner, there are several service tiles for 'New in DigiLocker', including 'Central Board of Secondary Education Class X, 2022 Marksheet', 'Food, Civil Supplies and Consumer Protection Department, Maharashtra Ration Card', 'Registration Department, Tamil Nadu Marriage, Death and Birth Certificates', and 'Universal Sampo Insurance Co. Ltd General Insurance Policies Certificates'. A navigation menu on the left includes 'Home', 'Issued Documents', 'Search Documents', 'Drive', 'DigiLocker Services', 'My Profile', 'Activities', 'About DigiLocker', and 'Logout'. The bottom screenshot shows the 'ICAR AIEEA (UG) Confirmation Page' with a form to 'Get your document by entering the required details'. The form fields are: 'Name (as per Aadhaar)', 'Date of Birth', and 'Application No' (with an example 'Ex. 22XXXXXXXXXX'). There is a checkbox for 'I provide my consent to DigiLocker to share my details with the Issuers for the purpose of fetching my documents.' and a 'Get Document' button. A yellow box on the right contains text: 'National Testing Agency (NTA) conducts entrance examinations for admissions/fellowship in higher educational institutions. Related academic awards of 2022 are currently available in DigiLocker.'

# Digital Services

## Online Banking and BHIM App (<https://www.bhimupi.org.in/>)



# Public Utility portals of Government of India

ABDM (<https://abdm.gov.in/>)

The screenshot displays the ABDM website interface. At the top, there is a navigation bar with the National Health Authority logo, the Digital Mission logo, and a menu with options: HOME, ABOUT, PUBLICATIONS, MEDIA, TENDERS, OUR PARTNERS, CAREERS, FAQ, and SUPPORT. Below the navigation bar, a section titled "ABDM Components" features five icons representing: ABHA Number, Health Facility Registry, ABHA App, Healthcare Professionals Registry, and Unified Health Interface (UHI). A vertical button on the right side says "Download ABHA App".

Below the components, a text block explains the ABHA Number: "Your Ayushman Bharat Health Account (ABHA) Number is a hassle-free method of accessing and sharing your health records digitally. It enables your interaction with participating healthcare providers, and allows you to receive your digital lab reports, prescriptions and diagnosis seamlessly from verified healthcare professionals and health service providers".

An orange button labeled "Create ABHA Number" is positioned below the text. To the right, a hand is shown holding a sample ABHA card for a user named Suresh. The card displays the following information:

- Name: Suresh
- ABHA Number: XX-XXXX-XXXX-XXXX
- ABHA Address: XXXXX@abdm
- Date of Birth: 04/05/1990
- Gender: Male
- Mobile: +91 97112XXXX

A QR code is also visible on the card.



# Digital Infrastructure for Knowledge Sharing (DIKSHA) (<https://diksha.gov.in/>)

The screenshot displays the DIKSHA website interface. At the top, there is a navigation bar with the text "Skip to Main Content | Screen Reader Access | Site Map" and accessibility icons (A-, A, A+, and a speaker icon). The main header includes the Government of India logo, the DIKSHA logo, and navigation links for "Home", "Digital textbooks", "Courses", "TV classes", and "All". A search bar is present with the text "Search or enter QR code" and a "Search" button. To the right of the search bar, it says "Select language : English". Below the search bar, there are filters for "CBSE/NCERT" and various languages: "Bengali Medium", "English Medium" (highlighted), "Gujarati Medium", "Hindi Medium", "Kannada Medium", "Malayalam Medium", "Marathi Medium", "Odia Medium", and "Punjab". Further down, there are filters for "Class 1" through "Class 12" (highlighted), "CPD", and "Others". Below these, there are filters for "1 {FrameworkCategory4}" and "1 User Type", along with a "Reset" button. The main content area is titled "Digital textbooks" and shows a list of books under the "Hindi" category. Three books are visible: 1. "अंतरा (NEW) अंतरा भाग-2" (Hindi • Class 12), Medium: Hindi, English, CBSE. 2. "आरोह (NEW) आरोह भाग-2" (Hindi • Class 12), Medium: Hindi, English, CBSE. 3. "रिमझिम 3 (NEW) रिमझिम 3" (Hindi • Class 3), Medium: Hindi, English, CBSE. In the bottom right corner, there is a blue chat bubble with the text "Ask Tara" and a blue robot icon.

# Study Webs of Active Learning for Young Aspiring Minds (SWAYAM) Portal (<https://swayam.gov.in/>)

The screenshot shows the SWAYAM portal homepage. At the top, there is a navigation bar with the SWAYAM logo on the left and links for 'About Swayam', 'All Courses', 'National Coordinators', and 'Local Chapters' on the right. Below the navigation bar, there is a 'COURSE CATALOG' dropdown menu, a search bar with the placeholder text 'Search Catalog...', and a red 'SIGN-IN / REGISTER' button. The main content area features a corkboard background with four yellow boxes containing the following text: 'Submit your credit transfer grievances', 'SWAYAM Courses in Regional languages', 'Let COVID-19 not Stop your Learning. Continue with SWAYAM', and 'Admit cards live for Exam of Jan 2022 Sem scheduled on 29-30 August 2022'. Below this, the 'NATIONAL COORDINATORS' section displays logos for AICTE, CEC, IGNOU, IIMB, NCERT, NIOS, NITTR, NPTEL, and UGC. The 'SWAYAM BENEFITS' section is partially visible at the bottom, showing the text 'Best-in-Class Instructors' and a green upward-pointing arrow button.

# National portal of India (<https://www.india.gov.in/>)

The screenshot shows the homepage of the National Portal of India. At the top, there is a navigation bar with the URL 'india.gov.in', a search bar, and links for 'Skip to main content', 'Sign In', and 'Register'. Below this is the main header featuring the national emblem and the text 'india.gov.in national portal of india'. A row of icons represents 'TOPICS', 'SERVICES', 'MY GOVERNMENT', 'PEOPLE GROUPS', and 'INDIA AT A GLANCE'. A blue banner contains a search bar and a 'MOST SEARCHED' section listing 'Birth Certificate', 'Driving Licence', and 'Pan card'. The main content area features a large banner for 'DEFEXPO 22 INDIA' (18-22 October 2022, Gandhinagar, Gujarat) with images of a tank, a jet, a helicopter, a ship, and a statue. Below the banner are three sections: 'News Highlights' with a 'News & Press Releases' sub-section listing meetings by the Raksha Mantri; 'Most Requested Information & Forms' with an 'Information' sub-section listing various government services; and 'Activities & Initiatives' featuring the '75 Azadi Ka Amrit Mahotsav' logo.

india.gov.in  
national portal of india

Search - Keyword, Phrase Search Advanced Search MOST SEARCHED Birth Certificate | Driving Licence | Pan card

defexpo 2022 India : The Emerging Defence Manufacturing Hub DEFEXPO 22 INDIA 18 - 22 OCTOBER, 2022 GANDHINAGAR Ministry of Defence 18<sup>TH</sup>-22<sup>ND</sup> OCTOBER, 2022 AT GANDHINAGAR, GUJARAT india.gov.in

News Highlights

News & Press Releases

- » Raksha Mantri Shri Rajnath Singh holds bilateral meetings with...
- » Defence Secretary conducts bilateral meetings ahead of DefExpo...

Most Requested Information & Forms

Information

- » Department of Women, Children, Disabled and Senior Citizens...
- » Indira Gandhi National Open University (IGNOU)
- » Information on senior citizens schemes in Karnataka

Activities & Initiatives

75 Azadi Ka Amrit Mahotsav initiative of the Government of

<https://defexpo.gov.in>

# Unit-2: Communication and Collaboration in the Cyberspace

Unit	Unit name	Week No	Topic	Teaching Resources
II	Communication and Collaboration in the Cyberspace	6	Electronic Communication: electronic mail, blogs, social media	<a href="https://static.vikaspedia.in/media/files_en/education/Digital%20Literacy/Email5.pdf">https://static.vikaspedia.in/media/files_en/education/Digital%20Literacy/Email5.pdf</a> <a href="https://www.nidirect.gov.uk/articles/email-internet-and-social-media">https://www.nidirect.gov.uk/articles/email-internet-and-social-media</a>
		7	Collaborative Digital platforms	<a href="https://en.wikipedia.org/wiki/Digital_collaboration">https://en.wikipedia.org/wiki/Digital_collaboration</a>
		8-9	Tools/platforms for online learning, Collaboration using file sharing, messaging, video conferencing	<a href="https://vikaspedia.in/education/interactive-resources/digital-learning-resources">https://vikaspedia.in/education/interactive-resources/digital-learning-resources</a> <a href="https://www.prodigygame.com/in-en/blog/virtual-learning-tools/">https://www.prodigygame.com/in-en/blog/virtual-learning-tools/</a> <a href="https://dexterredward.com/secure-collaboration/">https://dexterredward.com/secure-collaboration/</a>

# Unit-2: Communication and Collaboration in the Cyberspace



# Unit-2: Communication and Collaboration in the Cyberspace

- Electronic mail
  - Gmail
  - Outlook
  - Yahoo!
- Blogs
  - [www.wordpress.org](http://www.wordpress.org)
  - [www.blogger.com](http://www.blogger.com)
- Social media
  - Facebook
  - Twitter
  - Instagram
  - Snapchat
- Tools/platforms for online learning, Collaboration using file sharing, messaging, video conferencing
  - Zoom
  - MS Teams
  - Various Google products (<https://about.google/products/#all-products>)
  - Webex
  - JioMeet

# Unit-2: Communication and Collaboration in the Cyberspace

Staying safe while communicating and collaborating in the Cyberspace:

- For teachers
  - Where ever possible create separate accounts for demonstrations during classes
- For all
  - Log out of accounts on all the websites be it email account, your account on Digi locker, etc. when logged in through some system that is not yours
  - Be careful about the content that you post / share on the online media.
    - think about its consequences in future
    - data on the Internet becomes ‘immortal’
  - At the end of the lab class, delete all the important files (from recycle bin/trash also) that you have downloaded during the lab class

# Unit-3: Towards Safe and Secure Cyberspace

Unit	Unit name	Week No	Topic	Teaching Resources
III	Towards Safe and Secure Cyberspace	10	Online security and privacy	<a href="https://safety.google/security-privacy/">https://safety.google/security-privacy/</a>
		11	Threats in the digital world: Data breach and Cyber Attacks	<a href="https://www.csoonline.com/article/3541148/the-biggest-data-breaches-in-india.html">https://www.csoonline.com/article/3541148/the-biggest-data-breaches-in-india.html</a> <a href="https://www.secureworld.io/industry-news/top-10-data-breaches-of-all-time">https://www.secureworld.io/industry-news/top-10-data-breaches-of-all-time</a> <a href="https://onlinedegrees.und.edu/blog/types-of-cyber-security-threats/">https://onlinedegrees.und.edu/blog/types-of-cyber-security-threats/</a>
		12	Blockchain Technology	<a href="https://blockchain.gov.in/">https://blockchain.gov.in/</a> <a href="https://blockchain.gov.in/assets/docs/blockchain_informatics.pdf">https://blockchain.gov.in/assets/docs/blockchain_informatics.pdf</a> <a href="https://blockchain.gov.in/assets/docs/Whitepaper_30jan.pdf">https://blockchain.gov.in/assets/docs/Whitepaper_30jan.pdf</a>
		13	Security Initiatives by the Govt of India	<a href="https://safety.google/">https://safety.google/</a> <a href="https://www.aicte-india.org/CyberSecurity">https://www.aicte-india.org/CyberSecurity</a> <a href="https://www.niti.gov.in/sites/default/files/2019-07/CyberSecurityConclaveAtVigyanBhavanDelhi_1.pdf">https://www.niti.gov.in/sites/default/files/2019-07/CyberSecurityConclaveAtVigyanBhavanDelhi_1.pdf</a>



# Unit-3: Towards Safe and Secure Cyberspace

Online Security	Online Privacy
Protect Your Phone, Personal Information and Data From Hackers	Protect Your Privacy on Apps
Recognize, Remove, and Avoid Malware	Protect Your Privacy Online
Recover Your Hacked Email or Social Media Account	Secure Your Voice Assistant and Protect Your Privacy
Remove Your Personal Information Before You Get Rid of Your Phone and/or Computer	Stalking Apps: What To Know
Secure Your Home Wi-Fi Network and Safely Use Public Wi-Fi Networks	What To Do if You're the Target of Revenge Porn
Use Two-factor Authentication to Protect Your Accounts	What To Do if You're the Target of Revenge Porn

# Threats in the digital world

- **Data breach:** A data breach is a theft of data by a malicious actor. Motives for data breaches include crime (i.e. identity theft), a desire to embarrass an institution (e.g. Edward Snowden or the DNC hack), and espionage.
- **Cyber Attacks:** A cyber attack is an attack that is mounted against us (meaning our digital devices) by means of cyberspace.

# Recent Data breaches in India

Source: <https://www.secureworld.io/industry-news/top-10-data-breaches-of-all-time>

- Air India data breach highlights third-party risk, May 2021
- CAT burglar strikes again: 1,90,000 applicants' details leaked to dark web, May 2021
- Hacker delivers 180 million Domino's India pizza orders to dark web, April 2021
- Trading platform Upstox resets passwords after breach report, April 2021
- Police exam database with information on 500,000 candidates goes up for sale, February 2021
- COVID-19 test results of Indian patients leaked online, January 2021
- User data from Juspay for sale on dark web, January 2021
- BigBasket user data for sale online, October 2020
- Unacademy learns lesson about security, May 2020
- Hackers steal healthcare records of 6.8 million Indian citizens, August 2019
- Local search provider JustDial exposes data of 10 crore users, April 2019
- SBI data breach leaks account details of millions of customers, January 2019

# International Data breaches

Source:<https://www.secureworld.io/industry-news/top-10-data-breaches-of-all-time>

- Yahoo data breach (2013)
- First American Financial Corporation data breach (2019)
- Equifax data breach (2017)
- Marriott International data breach (2018)
- Adult FriendFinder Networks data breach (2016)
- Facebook data breach (2019)
- Target data breach (2013)
- U.S. Office of Personnel Management data breach (2015)
- MySpace data breach (2013)
- LinkedIn data breach (2012)
- Adobe data breach (2013)
- SolarWinds supply chain data breach (2020)

# Cyber Attacks Examples

- Malware– Malicious software to disrupt computers
- Viruses, worms, ...
- Theft of Intellectual Property or Data
- Hactivism– Cyber protests that are socially or politically motivated
- Mobile Devices and applications and their associated Cyber Attacks
- Social Engineering– Entice Users to click on malicious links
- Spear Phishing– Deceptive Communications (e-mails, texts, tweets)
- Domain Name System (DNS) Attacks
- Router Security– Border Gateway Protocol (BGP) Hijacking
- Denial of Service (DoS)– blocking access to websites

# Blockchain Technology (<https://blockchain.gov.in/>)

Source: [https://blockchain.gov.in/assets/docs/blockchain\\_informatics.pdf](https://blockchain.gov.in/assets/docs/blockchain_informatics.pdf)

- A business involves transactions and information exchange among various stakeholders. Since most of the existing systems are centralized, there is greater risk to security, and this necessitates a secure and shareable system to help stakeholders interoperate efficiently.
- Blockchain is a distributed system where transaction records are bundled in blocks and linked with previous ones.
- Transaction data within a block is secure because it is encrypted and digitally signed.
- Bitcoin network is a peer to peer payment network, and it is an application of blockchain technology.

# Security Initiatives by the Govt of India

- [cert-in.org.in](https://cert-in.org.in)



- <https://www.csk.gov.in/>



- <https://infosecawareness.in/>



▶ Twitter

▶ Instagram

▶ Facebook

▶ LinkedIn

▶ You Tube

▶ Public

<https://infosecawareness.in/>

## Steps taken by the Central Government to spread awareness on prevention of cyber crime

- i. MHA's has launched '@CyberDost' twitter handle to spread awareness on cybercrime prevention and tweeted over 1066 cyber safety tips through short videos, images and creatives. It has more than 3.64 lakh followers.
- ii. Radio campaign.
- iii. Sent more than 100 crore SMSs on cybercrime awareness to public.
- iv. Publicity at regular interval on prevention of cyber crime and cyber safety tips through various Videos/GIFs initiated on various platforms as under:

Twitter	<a href="https://twitter.com/Cyberdost">https://twitter.com/Cyberdost</a>
Facebook	<a href="https://www.facebook.com/CyberDostI4C">https://www.facebook.com/CyberDostI4C</a>
Instagram	<a href="https://www.instagram.com/cyberdosti4c">https://www.instagram.com/cyberdosti4c</a>
Telegram	<a href="https://t.me/cyberdosti4c">https://t.me/cyberdosti4c</a>

- v. MyGov has been engaged by the I4C for the publicity on the topic of prevention of cyber crimes through multiple mediums.
- vi. Published 'Handbook for adolescents/students on cyber safety'.
- vii. Published 'Information Security Best practices' for the benefit of Government Officials.
- viii. Organized of Cyber Safety and Security Awareness weeks through the C-DAC in association with Police Department in different States.
- ix. 148 Cyber Crimes Advisories have been shared by the I4C with States/UTs, Ministries/Departments as a preventive measure.
- x. Issued alerts/advisories to States/UTs from time to time.
- xi. Delhi Metro has been requested to publicize the National Cyber Crime Reporting Portal and National Toll-Free Helpline Number '1930'.
- xii. Two bilingual Manuals on "Cyber Hygiene for Cyber Space - Do's and Don'ts" (Basic and Advanced version) released in January, 2022 to provide basic cyber hygiene regarding internet safety, email, mobile safety, etc.
- xiii. States/UTs have been requested by the MHA to organize "Cyber Jaagrookta Diwas" on first Wednesday of every month at 11 am on cyber hygiene commencing from 6<sup>th</sup> October 2021 (Wednesday) onwards and launch mass awareness campaign in vernacular languages for all schools/colleges. Also requested to prepare 'Annual Action Plan' in this regard.
- xiv. The Ministry of Education has been requested to commence Curriculum in cyber security and cyber hygiene from 6<sup>th</sup> to 12<sup>th</sup> standards for all the streams, so as to provide basic awareness to all students in schools for all CBSE schools at Central/State/UT level.
- xv. Quarterly Newsletter of I4C (1<sup>st</sup> and 2<sup>nd</sup> Edition) launched in January, 2022 to share information to counter the menace of cyber crime to Law Enforcement Agencies and Policymakers. The Newsletter comprises of latest cyber crimes trends, cyber crime statistics, National/International developments relating to prevention of cyber crimes, etc.

\*\*\* <https://infosecawareness.in/>

## Governments Initiatives

▶ Cyber Crime Awareness Booklet on Cyber Security Awareness

▶ Cyber Hygiene for cyber space

▶ Handbook on Cyber Safety for Students

▶ UGC Public Notice regarding: Initiatives for promoting Cyber Hygiene/Cyber Security in HEIs

▶ Steps taken by Central Governments to spread awareness on prevention of Cyber Crime

▶ Photo Gallery

▶ Radio Jingles

<https://infosecawareness.in/>



# Unit-4: Ethical Issues in Digital World

<b>Unit</b>	<b>Unit name</b>	<b>Week No</b>	<b>Topic</b>	<b>Teaching Resources</b>
<b>IV</b>	Ethical Issues in Digital World	14	Netiquettes	<a href="https://www.kaspersky.com/resource-center/preemptive-safety/what-is-netiquette">https://www.kaspersky.com/resource-center/preemptive-safety/what-is-netiquette</a> <a href="https://www.britannica.com/topic/netiquette">https://www.britannica.com/topic/netiquette</a>
		15	Ethics in digital communication, Ethics in Cyberspace	<a href="https://onlinelibrary.wiley.com/doi/book/10.1002/9781118968888">https://onlinelibrary.wiley.com/doi/book/10.1002/9781118968888</a> <a href="https://iccopr.com/wp-content/uploads/2017/10/Ethics-in-Digital-Communications-Guidelines.pdf">https://iccopr.com/wp-content/uploads/2017/10/Ethics-in-Digital-Communications-Guidelines.pdf</a> <a href="https://link.springer.com/chapter/10.1007/978-90-481-2370-4_1">https://link.springer.com/chapter/10.1007/978-90-481-2370-4_1</a>

# Unit-4: Ethical Issues in Digital World

## NETIQUETTES



**Communication  
Etiquettes**



**Social Media  
Etiquettes**



**Internet  
Etiquettes**

# Ethics in digital communication

1. Contribute to society and human being
2. Avoid harm to others
3. Be honest and trustworthy
4. Be fair and take action not to discriminate
5. Honor property rights including copyrights and patents
6. Give proper credit for intellectual property
7. Respect the privacy of others
8. Honor confidentiality

# Ethics in Cyberspace

- Do not use a computer to harm other people.
- Do not interfere with other people's computer work.
- Do not snoop around in other people's files.
- Do not use a computer to steal.
- Do not use a computer to bear false witness.
- Do not use or copy software for which you have not paid.
- Do not use other people's computer resources without authorization.
- Do not appropriate other people's intellectual output.
- Do think about the social consequences of the program you write.
- Do use a computer in ways that show consideration and respect.

# Suggestive list of Practicals / Activities

1. Identify few people of different age groups among family members, domestic help, neighborhood etc. Find out digital literacy among these groups. For example Usage of Smart phones, Internet, computer systems, and awareness of accessing an app.
2. Access your Digi Locker account. Fetch and use any three important documents from the Digi Locker such as class 10th certificate, 12th certificate, Aadhaar card, Driving License and vaccination certificate etc.
3. Students will explore digital services in the areas such as e-Kranti: education, health, planning, farming, security, financial inclusion, and justice. (any one site)
4. Explore the Digital India website (<https://www.digitalindia.gov.in>). List down any four achievements of the government towards the realization of the Digital India vision.
5. Prepare 2-3 page report on 9-pillars of Digital India.

# Suggestive list of Practicals / Activities (Cont.)

6. Create an email account, if you don't have one and explore it. Perform the following activities on your existing / new email account:
  - Create a group of your friends and post a welcome message on it.
  - Share a google document with your friends and give edit rights to a few of them.
  - Make changes to the account settings such that every outgoing email has your name, designation (if any) and your contact number.
  - Draft a mail and schedule it to be sent to some acquaintance the next day at 11am.
  - If you cannot attend to the emails for some days due to some urgent work, then change the settings such that an automatic reply mail is sent to the sender during those days.
  
7. Create an account on Twitter, if you don't have one and explore it. Perform the following activities on your existing / new Twitter account:
  - Follow the official handles of your college (@ your\_college), University of Delhi (@UnivofDelhi), or any other relevant entity
  - Send a message to someone you are following
  - Retweet a message

# Suggestive list of Practicals / Activities (Cont.)

8. Create an account on Facebook, if you don't have one and explore it. Perform the following activities on your existing / new Facebook account:
  - Follow the official page of your college, University of Delhi, or any other relevant authority.
  - Manage your privacy to limit who can interact with you and what you share.
  - Control who can see what you share on Meta.
9. Write a blog about your experience on your first day in the college.
10. Identify at least three collaborative digital platforms that can be used for group study. List their advantages and disadvantages, if any. Out of these platforms, which one you would like to use and why? Prepare 2-3 slides and present them.
11. Schedule a video meeting with your friend in the digital platform of your choice. After joining the meeting, send message, chat, and share a file.

# Suggestive list of Practicals / Activities (Cont.)

12. Students need to explore various blogs/ social media. Students may share their experiences through 10-15 minutes audio-visual presentation.
13. Students need to explore any one of the digital platform related to financial or banking or education sector.
14. Create your account in the NPTEL/ SWAYAM education portal. Enroll in any course of your choice.
15. Prepare a small report on either a data breach or a cyber attack.



# Suggestive list of Practicals / Activities (Cont.)

16. Browse the Govt blockchain website "[blockchain.gov.in](https://blockchain.gov.in)". Explore and present any one case study present on the website.
17. Explore the site "<https://www.csk.gov.in/>".
18. When students' online behavior draws criticism from others, what kind of guidelines help us as a society determine what is acceptable in cyberspace?
19. Do important people in your life talk with you about the dangers of certain online behavior?
20. Do young people fully understand the consequences of their online activity? Why or why not?