Department of Computer Science
University of Delhi

Model Curricula under UGCF
June 2022

TABLE-6

Illustration - 1: Sample UGCF for Multidisciplinary Courses of Study @#

Semester	Core (DSC)	Elective (DSE)	Generic Elective (GE)	Ability Enhancement Course (AEC)	Skill Enhancement Course (SEC)	Internship/ Apprenticeship/Project (2)	Value addition course (VAC)	Total Credits
I	Discipline AI- (4) Programming fundamentals using C++		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
	Discipline BI- (4)							
	Discipline CI- (4)							
II	Discipline A2 (4) Data Structures		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
	Discipline B2 (4)							
	Discipline C 2 (4)							
	Students on exit shall be awarded Undergraduate Certificate (in the field of Multidisciplinary Study) after securing the requisite 44 credits in Semesters 1 and 11							Total= 44
III	Discipline A3 (4) Computer System Architecture	Choose from pool of courses, DSE A/B/C (4) OR Choose from pool of courses, GE -3 (4)		Choose one from a pool of AEC courses (2)	Choose one SEC OR Internship/ Apprenticeship/ Project/community Outreach (2)		Choose one from a pool of courses (2)	22 credits
	Discipline B 3 (4)							
	Discipline C 3 (4)							
IV	Discipline A4 (4) Operating Systems	Choose from pool of courses, DSE A/ B/C (4) OR in the alternative GE-4 (4)		Choose one from a pool of AEC courses (2)	Choose one SEC OR 'Internship/ Appr enticeship / Project/community outreach (2)		Choose one from a pool of courses (2)	22 Credits
	Discipline B 4 (4)							
	Discipline C 4 (4)							
	Students on exit shall be awarded Undergraduate Diploma (in the Field of Multidisciplinary Study) after securing the requisite BB credits on completion of Semester N							Total= 88

V	Discipline A5 (4) Database Management Systems Discipline B 5 (4) Discipline C 5 (4)	Choose one from a pool of courses DSE A/ B/ C - (4)	Choose one from a pool of courses GE-5 (4)		Choose one SEC OR Internship/ Appren tic eship / Project / Com m unity outreach (2)		22 credits
VI	Discipline A6 (4) Computer Networks Discipline B 6 (4) Discipline C 6 (4)	Choose one from a pool of courses DSE A/ B/ C - (4)	Choose one from a pool of courses GE-6 (4)		Choose one SEC OR ' In tern - ship/ App ren ticeship / Project / Research / Com m unity Outreach (2)		22 credits
	Students on exit shall be awarded Bachelor of (in the field of Multidisciplinary Study) after securing the requisite 1.32 credits on completion of Semester VI						Total= 132
VII	DSC-(4) Design and Analysis of Algorithms	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)				Dissertation on Major (4+2) OR Dissertation on Minor (4+2) OR Academic project/ Entrepreneurship (4+2)	22 credits

VIII	DSC- (4) Information security	Choose three DSE(3x4) courses OR Choose two D SE-(2x4) one GE (4) course OR Choose one DSE (4) and two GE (2x4) courses (total= 12)	Course (AEC)	Course (SEC)	ship / Project (2)	Dissertation on Major (4+2) **OR** Dissertation on Minor (4+2) **OR** Academic project/ Entrepreneurship (4+2)	22 credits
Students on exjt shall be awarded Bachelor of (in the Field of Multidisciplinary Study) (Honours or Honours with Academic Projects/Entrepreneurship) after securing the requisite 176 credits on completion of Semester VIII							Total = 176

Department of Computer Science

University of Delhi

Curriculum

Multidisciplinary Courses of Study

with Three Core Disciplines

under UGCF 2022

Approved in
UG Committee meeting held on May 4, 2022
Faculty of Mathematical Sciences meeting held on May 25, 2022

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 multidisciplinary four year undergraduate program under NEP, called as UGCF, with computer science as one of the disciplines, 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 program aims at integrating computational knowledge with other science streams to enable students for multidisciplinary academic and professional pursuits.The curriculum 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 and knowledge of data structures helps in writing more organized and efficient codes, it is proposed to have one course in programming and one in data structures early on so that the students can develop good programming skills in the first year. Students are familiarized with the hardware of computers in their third semester in case they choose to exit after the second year.

Employability: The course builds on a meaningful blend of knowledge across disciplines with computer science. Industry sectors across the globe are harnessing the power of computers to perform computationally intensive work in varied fields such as medicine, physics, finance. The CS curriculum for MCS aims at providing a holistic approach towards making its students ready to meet the computational needs of industry. Discipline specific cores which include courses in programming, data structures, computer architecture, operating systems , computer networks, analysis and design of algorithms and information security equip the students with a strong foundation of various concepts in computer science. This will help them in applying their knowledge of CS to other disciplines of their interests.

The curriculum also aims to equip the students with tools and techniques of Artificial Intelligence by way of pathways in Machine Learning, Data Science.

Flexibility to Choose: The curriculum provides multiple pathways for the students to choose their micro-specialization, if any. Various tracks on Data-Science, Machine Learning and its application, Internet Technologies, Computer graphics,Security, networks and Theory encourage the students to explore domains of their choice. Each pathway provides learning progression in the chosen area of specialization. Students will be expected to model real life problems mathematically and use their programming skills to solve them.

Research: The course recognizes the essential need of research for building a strong society and a nation. With the knowledge gained in three core areas, a student of the multidisciplinary course of study can follow a proactive role in research, thus contributing significantly to integrative research areas. The course provides a strong foundation of IT application in different areas such as research in biology, physics, chemistry.mathematics.The curriculum motivates the students to pursue research in the area of their interest. The core knowledge imparted in the essential domain and the choice of in-demand elective papers, provides the requisite foundation for pursuing research in an area of interest.

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. 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	DSGA Contributed to	Desirable but not Prerequisite
I	DSC 01 (A1)	Programming fundamentals using C++	3	0	1	4	DSGA1	Nil
II	DSC 02 (A2)	Data structures	3	0	1	4	DSGA1, DSGA2	DSC 01/a course in C/C++ at plus 2 level/**
III	DSC 03 (A3)	Computer System Architecture	3	0	1	4	DSGA3	Nil
IV	DSC 04 (A4)	Operating systems	3	0	1	4	DSGA3	DSC 01/a course in C/C++ at plus 2 level/**
V	DSC 05 (A5)	Database Management systems	3	0	1	4	DSGA3	DSC 01/a course in C/C++ at plus 2 level/** DSC 04
VI	DSC 06 (A6)	Computer Networks	3	0	1	4	DSGA3	DSC 01/a course in C/C++ at plus 2 level/** DSC 02 DSC 04
VII	DSC 07	Design and Analysis of Algorithms	3	1	0	4	DSGA2	DSC 01/a course in C/C++ at plus 2 level/**, DSC 02
VIII	DSC 08	Information Security	3	0	1	4	DSGA1, DSGA4	DSC 01 DSC 02, DSC 03 DSC 04 DSC 05 DSC 06 DSC 07

Note:

1. **Batch size for Practicals will be (8-10) and Tutorials will be (12-15).**
2. **Wherever DCS01 is a prerequisite, a course in C/C++ 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 (A1): Programming Fundamentals 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 spanning multiple domains.

Course Learning Outcomes

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

- Write simple programs using built-in data types of C++.
- Implement arrays and user defined functions in C++.
- Write programs using dynamic memory allocation, handling external files, interrupts and exceptions.
- Solve problems spanning multiple domains using suitable programming constructs in C++.
- 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, 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, File I/O Basics, File Operations

References

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

Additional References

(i) Herbert Schildt, C++: *The Complete Reference*, 4th edition, McGraw Hill, 2003.

(ii) A. B. Forouzan, Richard F. Gilberg, Computer Science: *A Structured Approach using C++,* 2nd edition, Cengage Learning, 2010.

Suggested Practical list

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

$$S = 1 - 2^n + 3^n - 4^n + \ldots$$

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:

```
A
BA
CBA
DCBA
```

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

3. Write a program to compute the factors of a given number using default argument.
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 search
   c. Search an element in the array using binary search (both iterative and recursive versions)
   d. Display the address of every element of the array
5. Write a menu driven program to perform the following operations on a string:
   a. Calculate length of the string (use pointers)
   b. Check whether the first character of every word in the string is in uppercase or not
   c. Reverse the string
   d. Display the address of every character in 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.
8. Copy the contents of one text file to another file and display the number of characters copied.

DSC 02 (A2): Data Structures

Course Objective

The course aims at developing the ability to define, differentiate, implement the basic data structures like arrays, stacks, queues, lists, trees and use them 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 Recursion: Recursive functions, linear recursion, binary recursion.

Unit 4 Trees, Binary Trees: Trees: definition and properties, 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 (by copying), search operations, time complexity analysis of these operations; Balanced Search Trees: motivation and introduction.

Unit 6 Binary Heap, Priority Queue: Binary Heaps: motivation and introduction, application of heaps - Priority Queues (introduction only).

References

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

Additional References

(i) Sahni, S., *Data Structures, Algorithms and applications in C++*, 2nd 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. Perform matrix addition and multiplication.
2. Implement following recursive functions:
    a. Factorial of a number
    b. Nth 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.

DSC 03 (A3): 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 Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
2. Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
3. Determine various stages of the instruction cycle and describe interrupts and their handling.
4. Explain how the CPU communicates with memory and I/O devices.
5. Simulate the design of a basic computer using a software tool.

Syllabus

Unit 1 Digital Logic Circuits: Digital Logic Gates, Flip flops and their characteristic table, Logic circuit simplification using Boolean algebra and Karnaugh map, Don't care conditions, Combinational circuits, Introduction to Sequential circuits.

Unit 2 Digital Components: Decoders, Encoders, Multiplexers, Binary Adder, Binary Adder Subtractor, Binary Incrementer, Registers and Memory units.

Unit 3 Data Representation: Binary representation of both numeric and alphanumeric data, representation of numeric data in different number systems, (Binary, Octal, Decimal and Hexadecimal), conversion from one number system to another, complements, representation of signed and unsigned numbers, addition and subtraction of signed and unsigned numbers and overflow detection.

Unit 4 Basic Computer Organization and Design: Stored program organization, Computer registers, Instruction set and their completeness, Instruction cycle, Memory reference instructions, Register reference instructions, Input- Output reference instructions, Interrupt cycle, Addressing modes.

Unit 5 Input-Output Organization: I/O interface, I/O vs. Memory Bus, Isolated I/O, Memory Mapped I/O, Direct Memory Access.

References

1. Mano, M. *Computer System Architecture*, 3rd edition, Pearson education,1992.

Additional References

(i) Mano, M., *Digital Design*, Pearson education Asia, 1995.

(ii) Null, L., & Lobur, J., *The Essentials of Computer Organization and Architecture*, 5th edition, (Reprint) Jones and Bartlett Learning, 2018.

(iii) Stallings, W., *Computer Organization and Architecture Designing for Performance* 8th 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:

Registers

IR	DR	AC	AR	PC	I	E
16 bits	16 bits	16 bits	12 bits	12 bits	1 bit	1 bit

Memory 4096 words 16 bits per word	Instruction format 15 12 11 0 Opcode / Address

Basic Computer Instructions

Memory Reference			Register Reference	
Symbol	Hex		Symbol	Hex
AND	0xxx		CLA	7800
ADD	1xxx		CLE	7400

LDA	2xxx	Direct Addressing	CMA	7200
STA	3xxx		CME	7100
			HLT	7001

Refer to Chapter-5 for description of instructions.

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

1. Implement fetch sequence
2. Write an assembly program to simulate addition of two numbers when one number is stored in memory and another is entered by the user.
3. Write an assembly program to simulate addition of two numbers when both numbers are taken as inputs from user.
4. Write an assembly program to simulate subtraction of two numbers when one number is stored in memory and another is entered by the user.
5. Write an assembly program to simulate subtraction of two numbers when both numbers are taken as inputs from user
6. Write an assembly program to simulate the following logical operations on two user-entered numbers.

    i.AND

    ii.OR

    iii.NOT

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. CLE

    ii. CLA

    iii. CMA

    iv. CME

DSC 04 (A4): 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*, 9th edition, John Wiley Publications, 2016.
2. Das, S., *Unix: Concepts and Applications*, 4th edition, TMH, 2009.

Additional References

(i) Dhamdhere, D. M., *Operating Systems: A Concept-based Approach,* 2nd 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*, 9th edition, Pearson Education, 2018.

(iv) Tanenbaum, A. S., *Modern Operating Systems.* 3rd 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
    - iii. Directory Management : cd, mkdir, rmdir, ls
2. Execute various LINUX commands for:
    - i. Process Control: fork, getpid, ps
    - 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 calculate sum of n numbers using Pthreads.
5. Write a program to generate a Fibonacci Series of numbers using Pthreads.
6. Write a program to implement best-fit and worst-fit allocation strategies
7. Write a program to copy files using system calls and using pthreads and compare timings.
8. Write a program to implement FCFS scheduling algorithm.
9. Write a program to implement SJF scheduling algorithm.
10. Write a program to implement non-preemptive priority based scheduling algorithm.

DSC 05 (A5): Database Management Systems

Course Objective

The course introduces the students to the fundamentals of database management system and its applications. Emphasis is given on the popular relational database system. Students will learn about the importance of database structure and its designing using Entity Relationship diagram and formal approach using normalization. Basic concepts of file indexing and transaction processing will be taught. The course would give students hands-on practice of structured query language to create, manipulate and implement a relational database.

Course Learning Outcomes

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 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 File indexing and Transaction Processing: Need of file indexes, types of indexes, file organizations, single- and multi-level indexing, concurrent execution of transactions, ACID properties, need of data recovery.

References

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

Additional References

(i) Connolly, T. M., Begg, C. E., *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th edition, Pearson, 2019.

(ii) Ramakrishnan, R., Gehrke, J., *Database Management Systems*, 3rd edition, McGraw-Hill, 2014.

(iii) Silberschatz, A., Korth, H.F., Sudarshan S., *Database System Concepts*, 7th 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	Roll No	StudentName	CourseID	DOB
	Char(6)	Varchar(20)	Varchar(10)	Date

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

ADMISSION	Roll No	CID	DateOfAdmission
	Char(6)	Char(6)	**Date**

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.

DSC 06 (A6): 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, Network Layer protocol- (ARP, IPV4, ICMP).

Unit 5 Transport and Application Layer: 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*, 5th edition, Pearson Education, 2012.

2. Forouzan, B. A.. *Data Communication and Networking*, 4th edition, McGraw-Hill Education, 2017.

Additional References

(i) Comer, D. E.. *Computer Networks and Internet,* 6th edition, Pearson education, 2015.

(ii) Stallings, W., *Data and Computer Communications,* 10th 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.

DSC 07: 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*. 4th edition. Prentice Hall of India. 2022.
2. Kleinberg, J., Tardos, E., *Algorithm Design,* 1st edition, Pearson, 2013.

Additional Reference

(i) Basse, S., Gelder, A. V., *Computer Algorithms: Introduction to Design and Analysis,* 3rd 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

DSC 08: 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.

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 User authentication and Access Control: Digital User Authentication Principle, Password-Based Authentication, Remote User Authentication, Security Issues for User Authentication Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Attribute-Based Access Control, Identity, Credential, and Access Management, Trust Frameworks.

Unit 4 Database and Data Center Security:
The Need for Database Security, SQL Injection Attacks, Database Access Control.

Unit 5 Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation — Infected Content - Viruses, Propagation — Vulnerability Exploit - Worms, Propagation — Social Engineering — 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. **Software Security Issues -** Handling Program Input, Writing Safe Program Code, Handling Program Input.

Unit 6 Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Public-Key Infrastructure.

Unit 7 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.

References

1. W. Stallings, L. Brown, *Computer Security: Principles and Practice*, 4th 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 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.

Department of Computer Science
University of Delhi

Generic Electives (GEs)

Multidisciplinary with Three Cores

Under UGCF 2022

Approved in
UG Committee meeting held on May17, 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

Semester	**GE -No.**	**Title**	**L**	**T***	**P***	**Total credits**	**Pre-requisites**
I	**GE1a**	Programming using C++	**3**	**0**	**1**	**4**	None
	GE1b	Programming with Python	**3**	**0**	**1**	**4**	None
II	**GE2a**	Data Analysis and Visualization using Python	**3**	**0**	**1**	**4**	A course in Python
	GE2b	Data Analysis and Visualization using Spreadsheet	**3**	**0**	**1**	**4**	None
III	**GE3a**	Database Management Systems					None
	GE3b	Java Programming					None

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

		Design and Development					
VIII	**GE8a**	Machine Learning	**3**	**1**	**0**	**4**	A course in probability, statistics, linear algebra and multivariate analysis
	GE8b	Digital Marketing and Social Media Analytics (under preparation)	**3**	**0**	**1**	**4**	Python and HTML
	GE8c	Introduction to Parallel Programming	**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*, 6th Edition, Pearson India, 2015.
2. E Balaguruswamy, *Object Oriented Programming with C++*, 8th edition, McGraw-Hill Education, 2020.
3. D.S. Malik, *C++ Programming: From Problem Analysis to Program Design*, 6th edition, Cengage Learning, 2013.

Additional References

(i) Herbert Schildt, C++: *The Complete Reference*, 4th edition, McGraw Hill, 2003.

(ii) A. B. Forouzan, Richard F. Gilberg, Computer Science: *A Structured Approach using C++,* 2nd 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 + … 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*, 2nd edition, McGraw Hill Education, 2018.

(ii) Guttag, J.V. *Introduction to computation and programming using Python,* 2nd 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 t2=(11,13,15) with t1.

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 matplot 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*. 2nd 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/, 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 10th 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. Interpret the plots and the results of the analysis

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*, 4th 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 increases the selling of more items and getting the profit of Rs. 50000/-. Use Goal Seek of 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 stats 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)

FULL NAME	GRADE 1/2/3	BASIC SALARY	HRA	PF	GROSS	NET	VA	VA>HRA

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?

9. You are required to prepare a payroll statement in the given format making maximum use of cellreferencing facility:

Code	**Name**	**Category**	**Is HRA to be Paid**	**Basic**	**DP**	**DA**	**HRA**	**TA**	**CCA**	**Gross**
1			Y							
2			N							
	Total									

Required:

- Basic salary (Allow any Basic salary in the range of Rs.10000-35000)
- DP is 50% of Basic Salary.
- DA (as a Percentage of Basic + DP) is more than 35000 then 40% of basic else 30% of basic.
- HRA is to be paid @ 40% of (Basic plus DP) to those whom HRA payable is yes.
- TA is to be paid @ Rs. 800 PM if Basic Salary is Less than Rs.12000, otherwise the TA is Rs. 1000 PM)
- CCA is to be paid @ Rs. 300 PM if Basic Salary is less than Rs.12000/- otherwise the CCA is Rs. 500 PM)
- Gross salary is the sum of Salary and all other allowances
- Deduction: a) GPF 10% of (Basic +DP) subject to a minimum of Rs.2000/- b) IT 10% of Gross Salary
- Net salary is Gross salary minus total deductions.

10. Consider the following worksheet for APS 1st year students:

S.No.	**Name**	**Physics**	**Chem**	**Bio**	**Maths**	**CS**	**Total**	**%**	**Grade**
1									
2									
3									

4									
5									

The value of Grade is calculated as follows:

If % >=90	Grade A
If % >=80 & <90	Grade B
If % >=70 & <80	Grade C
If % >=60 & <70	Grade D

Otherwise, students will be declared fail.

i) Calculate Grade using if function

ii) Sort the data according to total marks

iii) Apply filter to display the marks of the students having more than 65% marks.

iv) Enter the S.No. of a student and find out the Grade of the student using VLOOKUP.

v) Extract all records where name

a) Begins with “A”

b) Contains “A”

c) Ends with “A”

GE3a: Database Management Systems

Course Objective

The course introduces the students to the fundamentals of database management system and its applications. Emphasis is given on the popular relational database system. Students will learn about the importance of database structure and its designing using Entity Relationship diagram and formal approach using normalization. Basic concepts of file indexing and transaction processing will be taught. The course would give students hands-on practice of structured query language to create, manipulate and implement a relational database.

Course Learning Outcomes

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*, 7th Edition, Pearson Education, 2016.
2. Murach J., *Murach's MySQL,* 3th Edition, Pearson, 2019.

Additional References

(i) Connolly T. M., Begg C. E. *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th edition, Pearson, 2019.

(ii) Ramakrishnan R., Gehrke J. *Database Management Systems*, 3rd Edition, McGraw-Hill, 2014.

(iii) Silberschatz A., Korth H.F., Sudarshan S. *Database System Concepts*, 7th 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	**Roll No**	**StudentName**	**CourseID**	**DOB**
	Char(6)	Varchar(20)	Varchar(10)	Date

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

ADMISSION	**Roll No**	**CID**	**DateOfAdmission**
	Char(6)	Char(6)	**Date**

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* 7th edition, Addison-Wesley, 2013.

2. Herbert Schildt, *Java: The Complete Reference*, 10th edition, McGraw-Hill Education, 2018.

3. Cay S. Horstmann, *Core Java - Vol. I – Fundamentals*, 10th 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*, 3rd 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++,* 2nd edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C., Introduction to Algorithms. 4th edition. Prentice Hall of India. 2022.

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

Additional References

(i) Sahni, S. *Data Structures, Algorithms and applications in C++*, 2nd edition. Universities Press, 2011.

(ii) Langsam Y., Augenstein, M. J., & Tanenbaum, A. M. *Data Structures Using C and C++*, 2nd edition, Pearson, 2009.

Suggested Practical List

1. Perform matrix addition and multiplication.
2. Implement following recursive functions:
   a. Factorialof a number
   b. Nth 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,* 5th edition, Tata McGrawHill, 2010.
2. Minnick, J. ,*Web Design with HTML5 and CSS3*, 8th 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*, 4th edition, Mike Murach & Associates, 2018.

(ii) 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.
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*, 9th edition, John Wiley Publications, 2016.
2. Das, S., *Unix: Concepts and Applications*, 4th Edition, TMH, 2009.

Additional References

(i) Dhamdhere, D. M., *Operating Systems: A Concept-based Approach,* 2nd 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*, 9th edition, Pearson

Education, 2018.

(iv) Tanenbaum, A. S., *Modern Operating Systems.* 3rd 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*, 2nd 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, HHTP 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*, 12th edition, McGraw-Hill Education, 2021.
2. Hans Bergsten, *Java Server Pages*, 3rd edition, O'Reilly, 2003.
3. Jim Keogh, *The Complete Reference J2EE*, 1st 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*, *5th edition*, Pearson Education, 2012.

2. Forouzan, B. A., *Data Communication and Networking*, *4th edition*, McGraw-Hill Education, , 2017.

Additional References

(i) Comer, D. E., *Computer Networks and Internet,* 6th edition, Pearson Publication, 2015.

(ii) Stallings, W., *Data and Computer Communications*, 10th 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*, 2nd edition, Addison-Wesley, 2018.
2. D. Herron, *Node.js Web Development*, 5th edition, Packt Publishing, 2020.
3. A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, 1st edition, O'Reilly, 2017.
4. M. Masse, *REST API – Design Rulebook*, 1st 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, 4th edition, 2020.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3rd edition, Tata McGraw Hill, 2010.
3. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4th 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*, 5th 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^nb^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*, 4th 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 caeser 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*, 4th edition. Prentice Hall of India. 2022.
2. Kleinberg, J., Tardos, E., *Algorithm Design,* 1st edition, Pearson, 2013.

Additional References

(i) Basse, S., Gelder, A. V. *Computer Algorithms: Introduction to Design and Analysis.* 3rd 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*, 4th 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, TN, 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*, 2nd edition, Addison-Wesley, 2003.
2. Quinn, M., *Parallel Programming in C with MPI and OpenMP*, 1st 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

Department of Computer Science
University of Delhi

Skill Enhancement Courses (SECs)

Under UGCF 2022

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/VII	SEC01A	Front End Web Design and Development	**1**	**0**	**1**	**2**	Nil
	SEC01B	Office Automation Tools	**1**	**0**	**1**	**2**	Nil
	SEC01C	App Development for Android	**1**	**0**	**1**	**2**	Basic knowledge of programming
	SEC1D	Basics of Programming in Python	**1**	**0**	**1**	**2**	For non-CS students
II/IV/VI/VIII	SEC02A	Backend Web Development	**1**	**0**	**1**	**2**	Knowledge of HTML, CSS
	SEC02B	Data Analysis using Spreadsheet	**1**	**0**	**1**	**2**	Nil
	SEC02C	App Development for iOS	**1**	**0**	**1**	**2**	Basic knowledge of programming
	SEC2D	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*, 5th 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***,** 4th 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*, 4th 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,* 2nd edition, McGraw Hill Education, 2018.

Additional References

1. Brown, Martin C., *Python: The Complete Reference*, 2nd edition, McGraw Hill Education, 2018.
2. Guttag, J.V., *Introduction to computation and programming using Python,* 2nd 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. Societies

   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 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 5.1Writer, 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

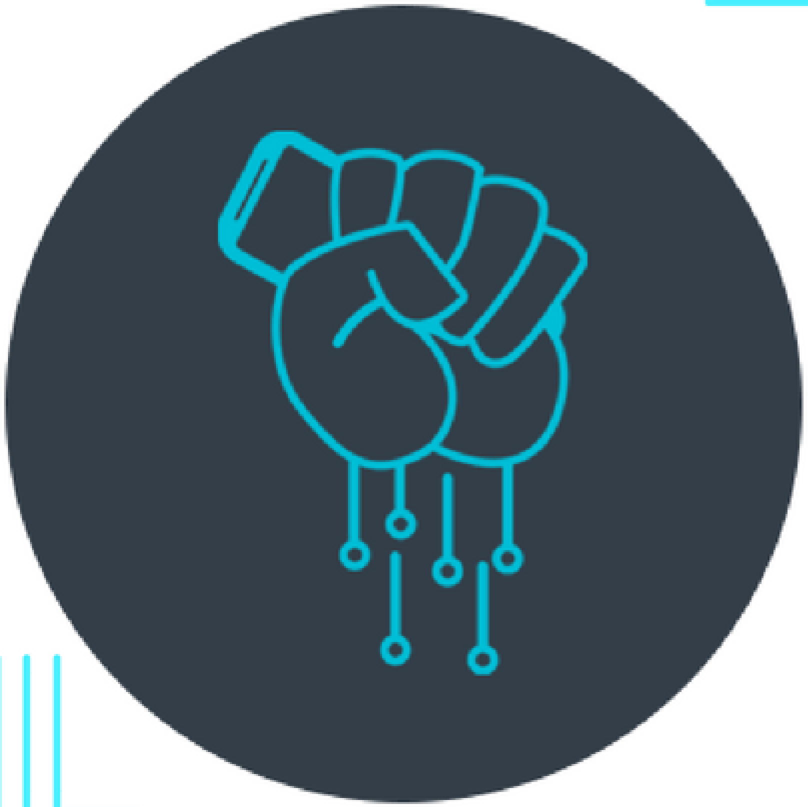
1. Brown, Martin C. *Python: The Complete Reference*, 2nd edition, McGraw Hill Education, 2018.
2. Guttag, J.V. *Introduction to computation and programming using Python,* 2nd edition. MIT Press, 2016.

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
Digital Empowerment	**02**	Lecture	Tutorial	Practical/ Practice	Pass in Class 12th	NIL
		1	**0**	**1**		

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	https://www.digitalindia.gov.in
		2-3	DigiLocker, E-Hospitals, e-Pathshala, BHIM, e-Kranti (Electronic Delivery of Services), e-Health Campaigns	https://digilocker.gov.in/ https://nha.gov.in/ https://pmjay.gov.in/ https://diksha.gov.in/ https://swayam.gov.in/ https://digitalindia.gov.in/content/ekranti
		4-5	Public utility portals of Govt. of India such as RTI, Health, Finance, Income Tax filing, Education	https://www.india.gov.in/ https://rtionline.gov.in/index.php https://www.bhimupi.org.in/ https://www.incometax.gov.in/iec/foportal https://nha.gov.in/ https://swayam.gov.in/ https://nptel.ac.in/

Unit 1: Digital Inclusion and Digital Empowerment

(Cont.)

Introduction to basic terminology:

- Digital world
- Digital citizen
- Digital inclusion
- Digital divide
- Digital empowerment

Need

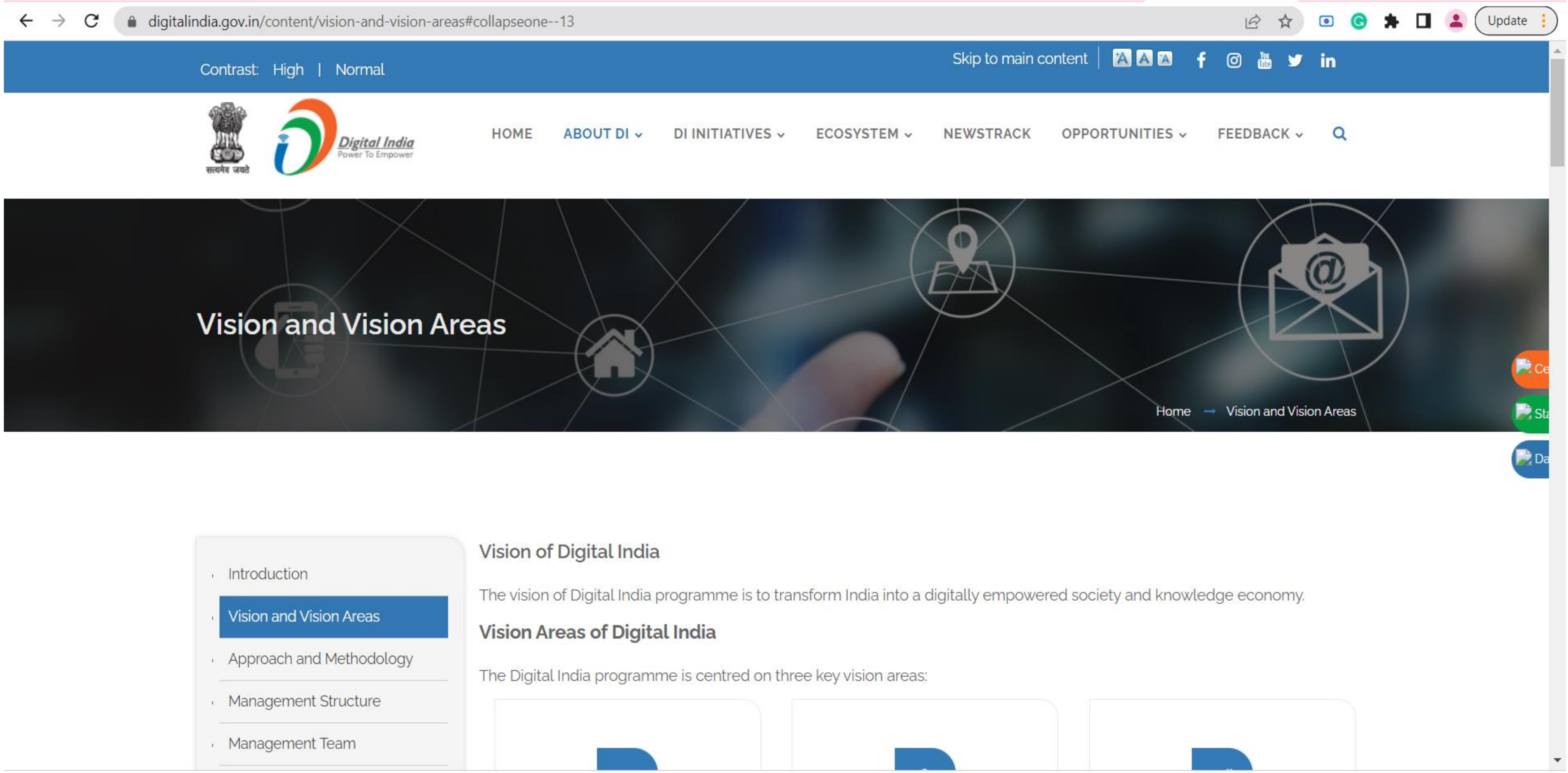
- Sustainable economy
- Healthcare
- Educational Achievements
- Efficient Civic Engagements

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.”*



Vision and Vision Areas

- Introduction
- Vision and Vision Areas
- Approach and Methodology
- Management Structure
- Management Team

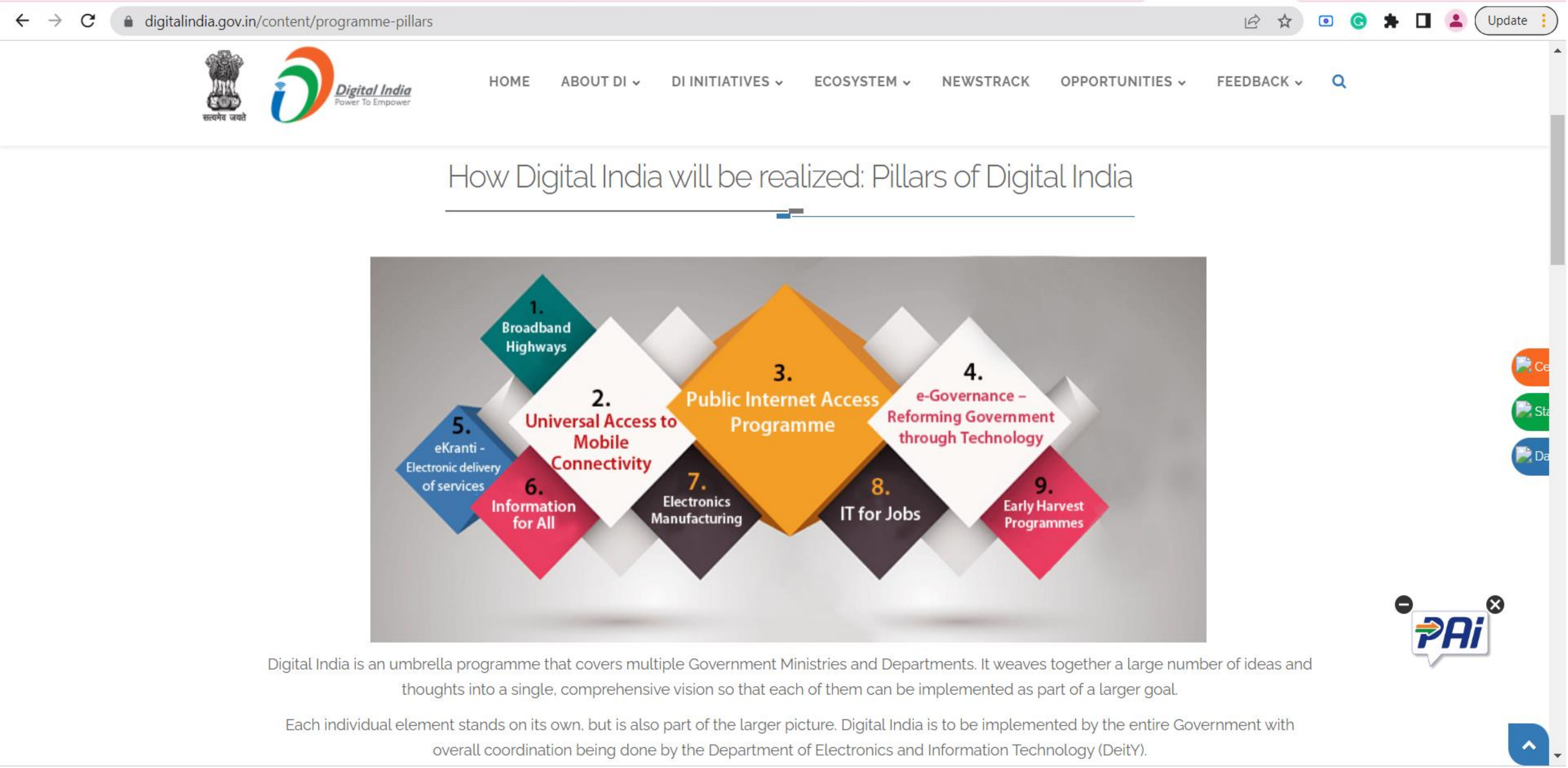
Vision of Digital India

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)



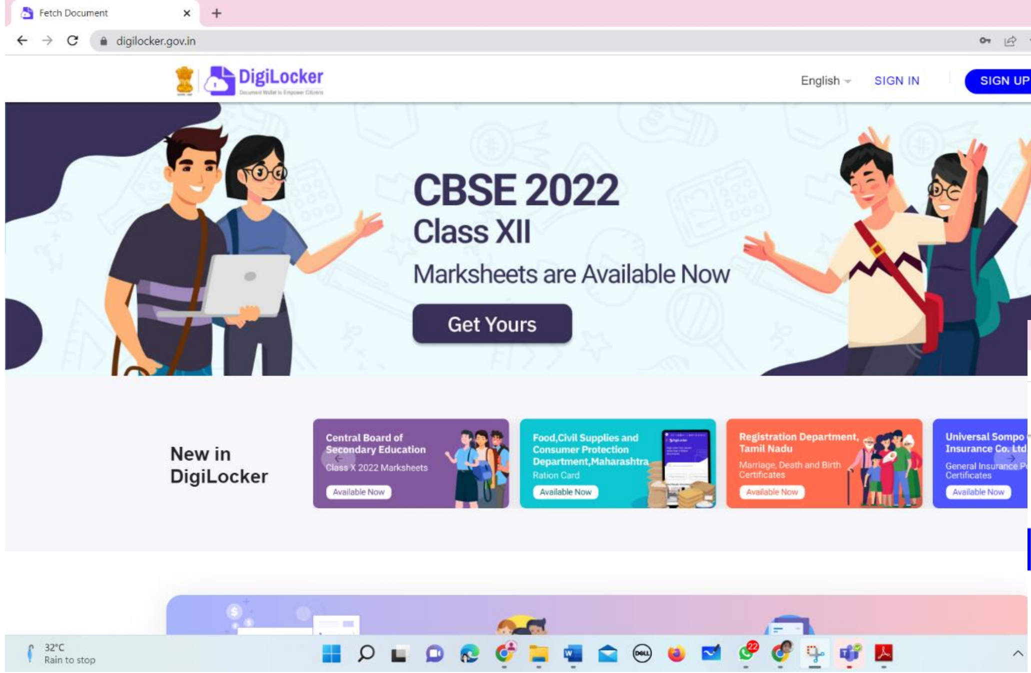
digitalindia.gov.in/content/programme-pillars
Update
सत्यमेव जयते
Digital India
Power To Empower
HOME
ABOUT DI
DI INITIATIVES
ECOSYSTEM
NEWSTRACK
OPPORTUNITIES
FEEDBACK
1. Broadband Highways
2. Universal Access to Mobile Connectivity
3. Public Internet Access Programme
4. e-Governance – Reforming Government through Technology
5. eKranti - Electronic delivery of services
6. Information for All
7. Electronics Manufacturing
8. IT for Jobs
9. Early Harvest Programmes
PAi

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/)



CBSE 2022
Class XII
Marksheets are Available Now

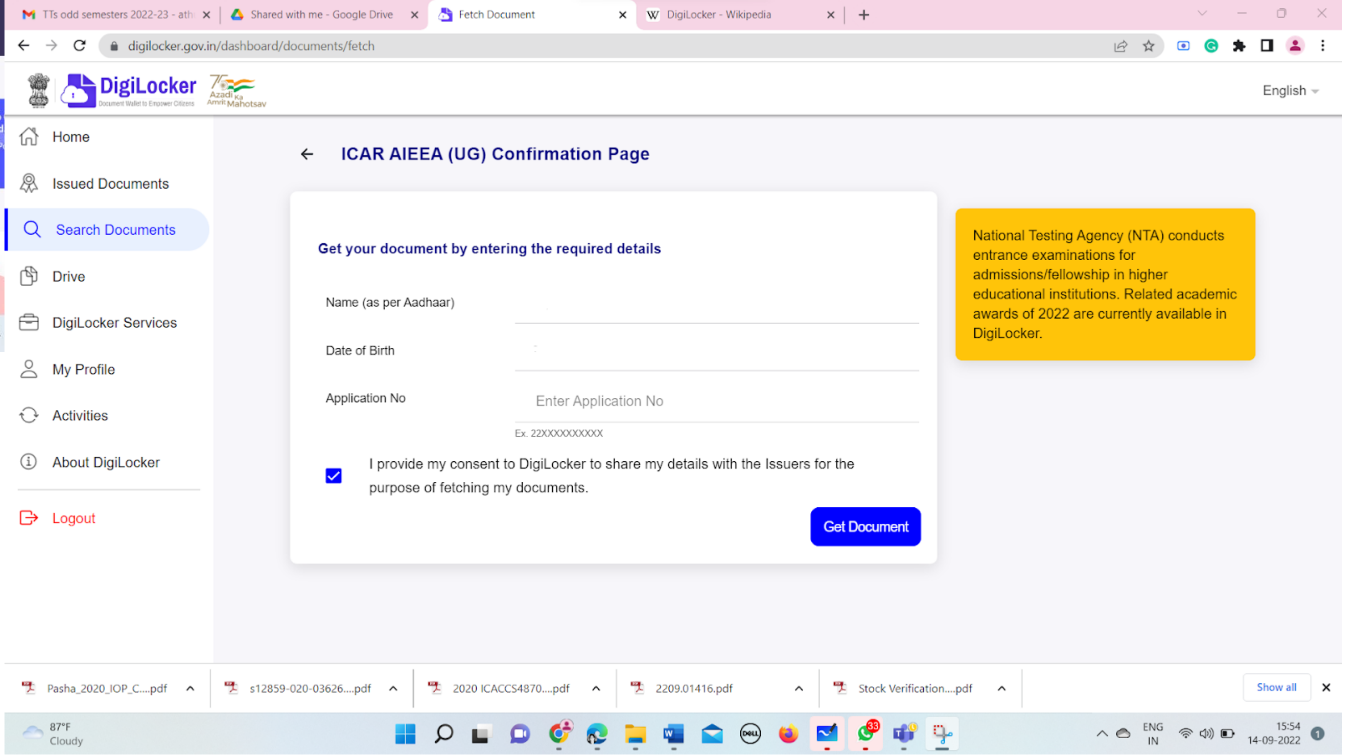
New in DigiLocker

Central Board of Secondary Education
Class X 2022 Marksheets
Available Now

Food,Civil Supplies and Consumer Protection Department,Maharashtra
Ration Card
Available Now

Registration Department, Tamil Nadu
Marriage, Death and Birth Certificates
Available Now

Universal Sompo Insurance Co. Ltd
General Insurance Certificates
Available Now



ICAR AIEEA (UG) Confirmation Page

Get your document by entering the required details

Name (as per Aadhaar)

Date of Birth

Application No: Enter Application No

Ex. 22XXXXXXXXXX

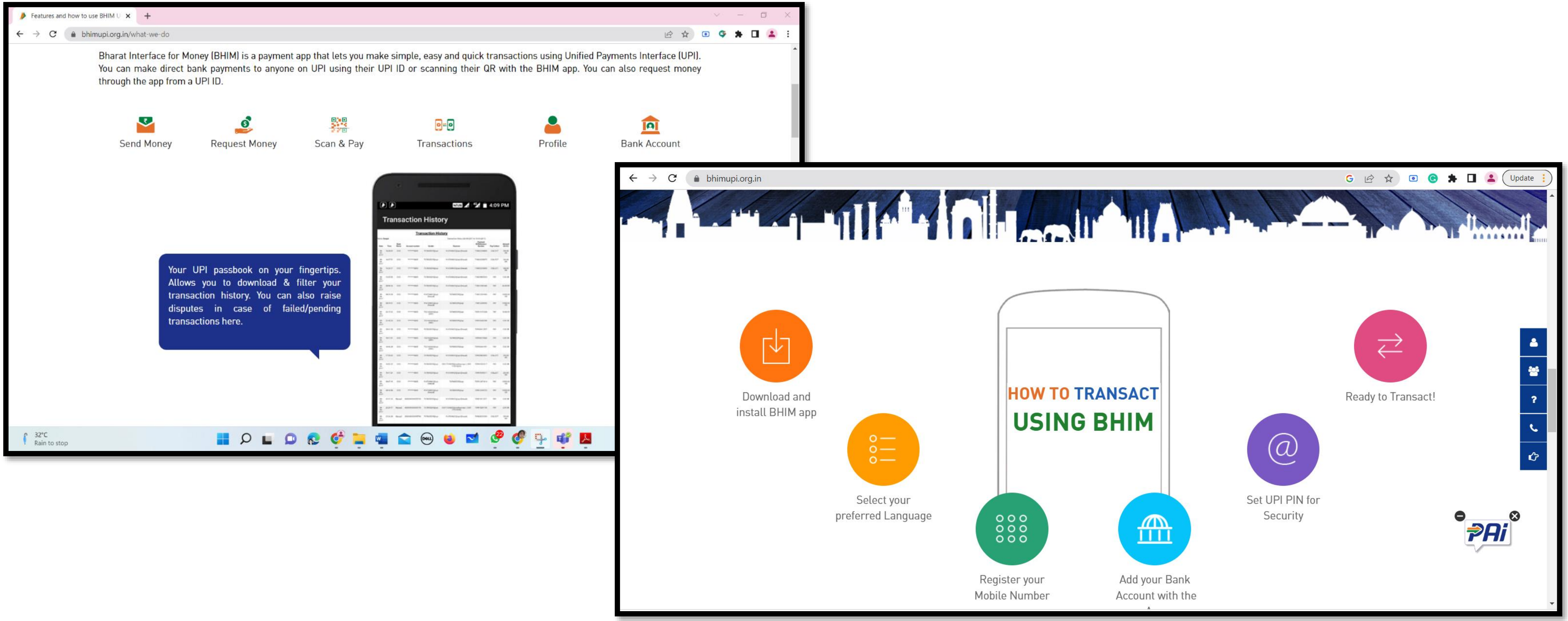
☑ I provide my consent to DigiLocker to share my details with the Issuers for the purpose of fetching my documents.

Get Document

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/)

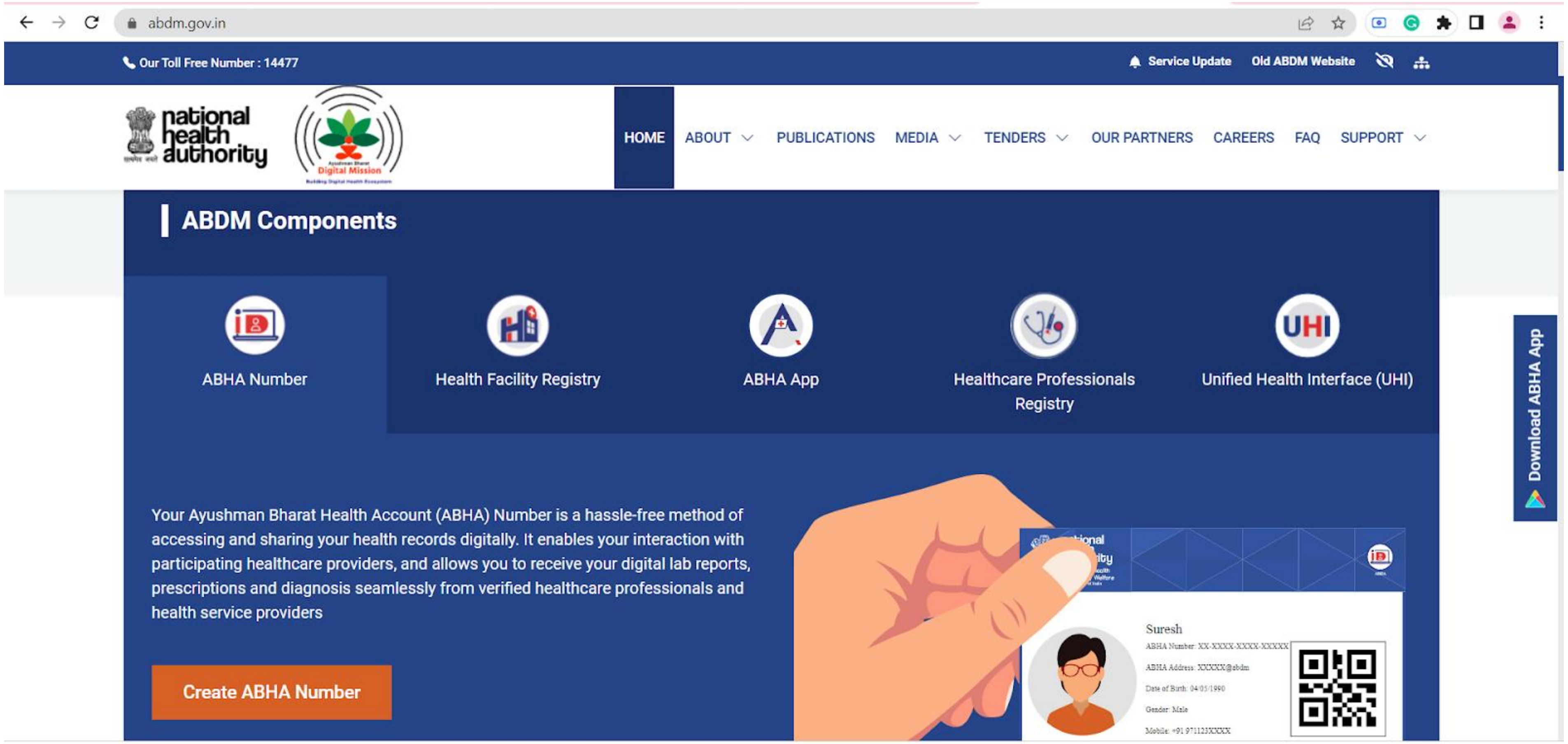


Bharat Interface for Money (BHIM) is a payment app that lets you make simple, easy and quick transactions using Unified Payments Interface (UPI). You can make direct bank payments to anyone on UPI using their UPI ID or scanning their QR with the BHIM app. You can also request money through the app from a UPI ID.

Your UPI passbook on your fingertips. Allows you to download & filter your transaction history. You can also raise disputes in case of failed/pending transactions here.

Public Utility portals of Government of India

ABDM (https://abdm.gov.in/)



ABDM Components

- ABHA Number
- Health Facility Registry
- ABHA App
- Healthcare Professionals Registry
- Unified Health Interface (UHI)

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

Create ABHA Number

Suresh

ABHA Number: XX-XXXX-XXXX-XXXXX

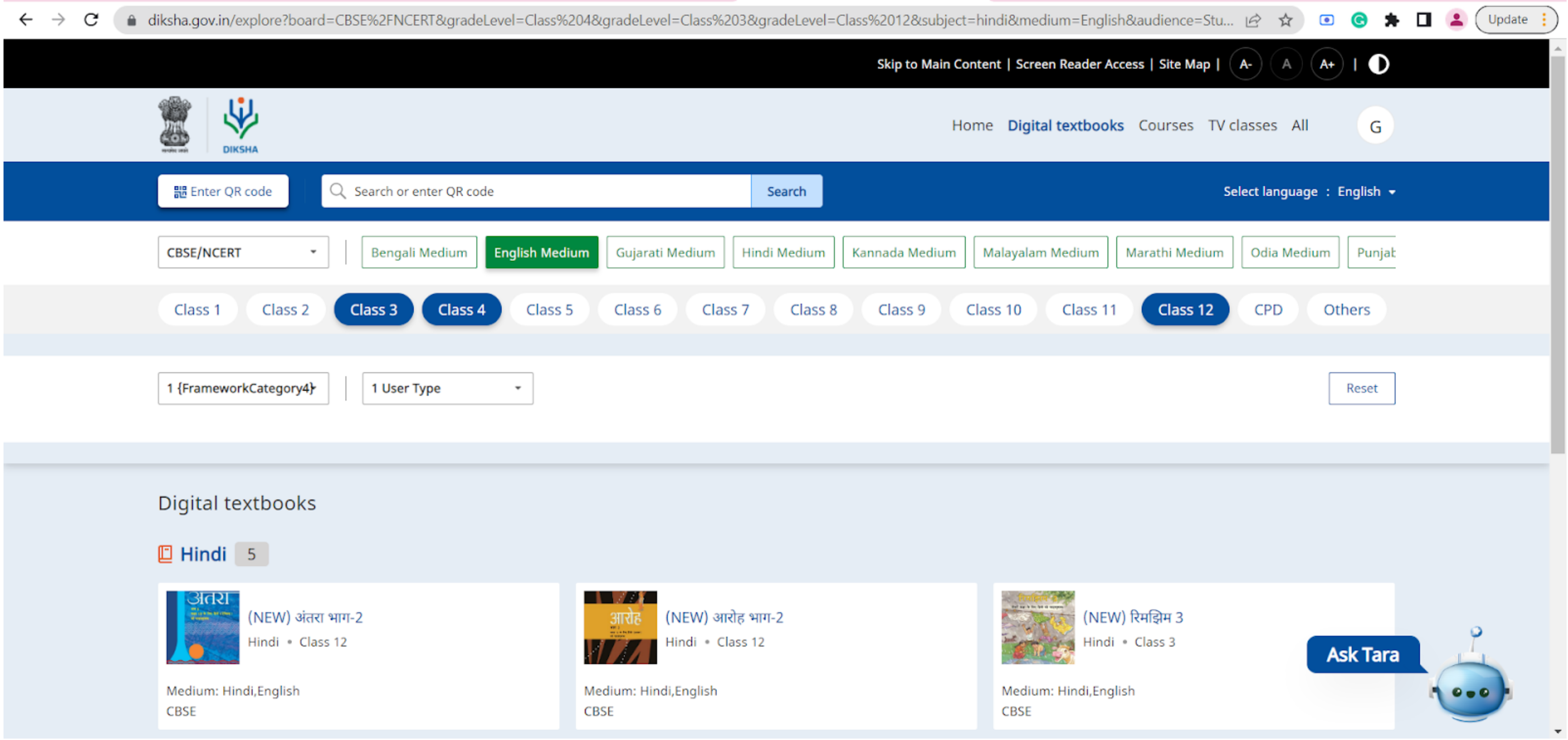
ABHA Address: XXXXX@abdm

Date of Birth: 04/05/1990

Gender: Male

Mobile: +91 971123XXXX

Digital Infrastructure for Knowledge Sharing (DIKSHA) (https://diksha.gov.in/)



diksha.gov.in/explore?board=CBSE%2FNCERT&gradeLevel=Class%204&gradeLevel=Class%203&gradeLevel=Class%2012&subject=hindi&medium=English&audience=Stu...

Digital textbooks

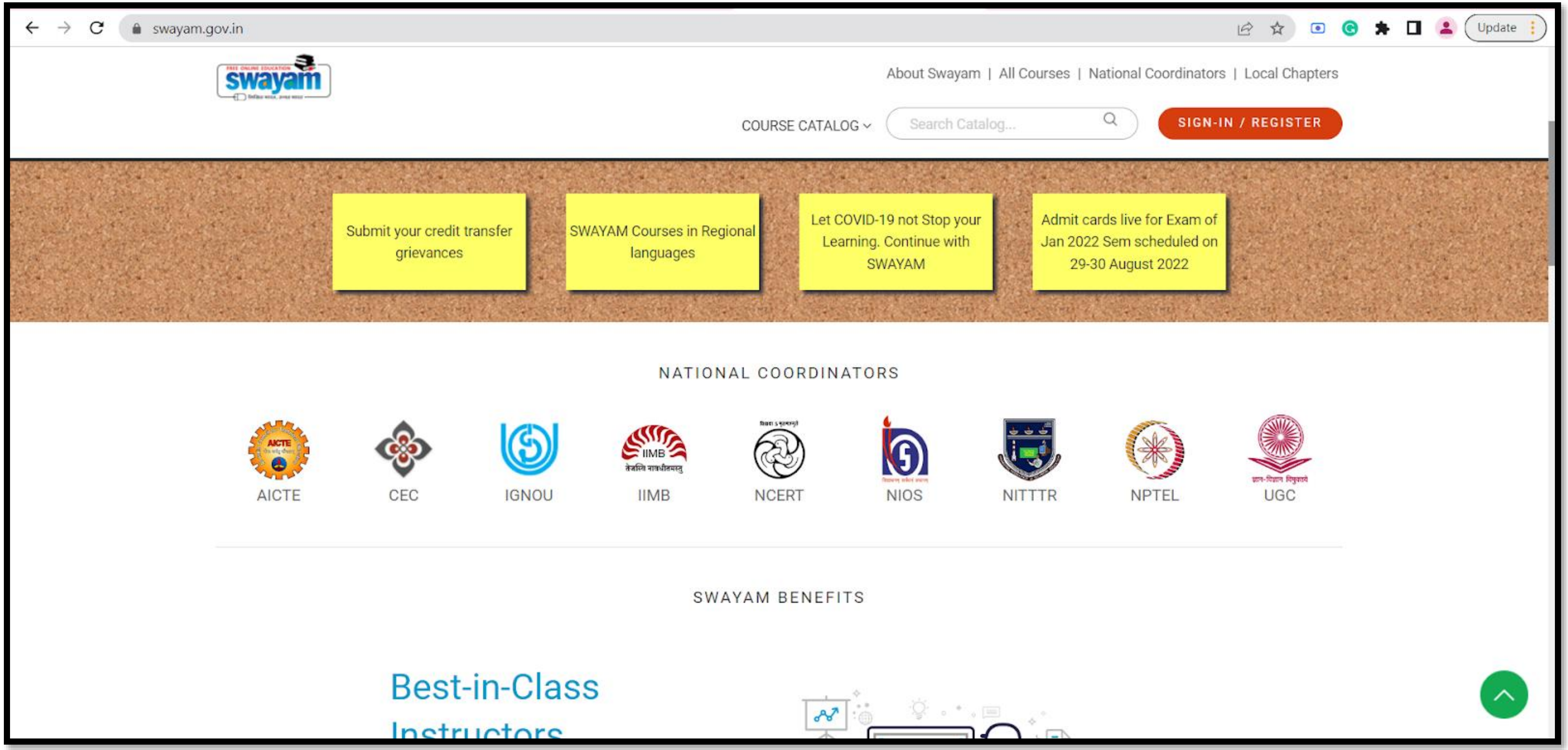
Hindi 5

(NEW) अंतरा भाग-2
Hindi • Class 12
Medium: Hindi,English
CBSE

(NEW) आरोह भाग-2
Hindi • Class 12
Medium: Hindi,English
CBSE

(NEW) रिमझिम 3
Hindi • Class 3
Medium: Hindi,English
CBSE

Study Webs of Active Learning for Young Aspiring Minds (SWAYAM) Portal (https://swayam.gov.in/)



Submit your credit transfer grievances

SWAYAM Courses in Regional languages

Let COVID-19 not Stop your Learning. Continue with SWAYAM

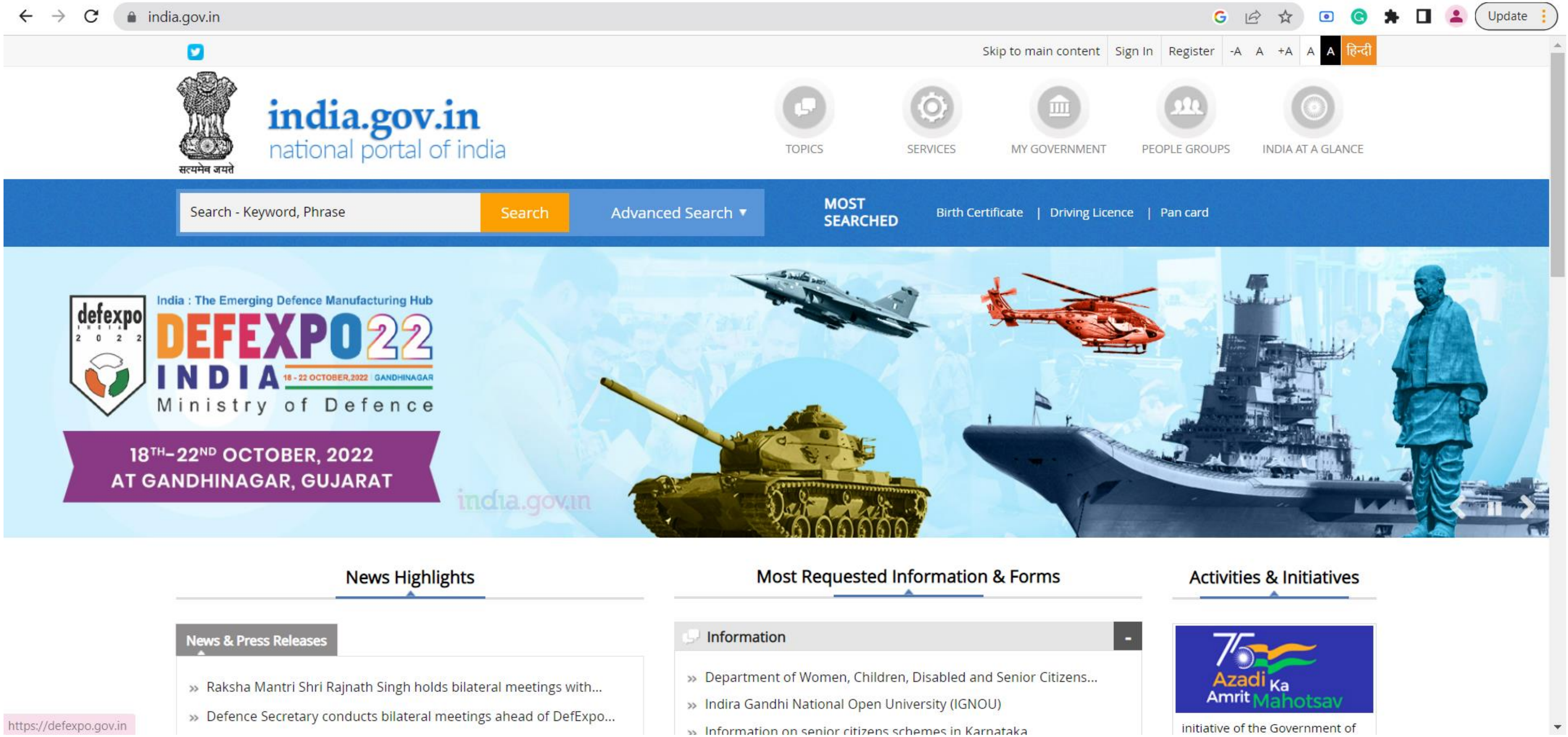
Admit cards live for Exam of Jan 2022 Sem scheduled on 29-30 August 2022

NATIONAL COORDINATORS

SWAYAM BENEFITS

Best-in-Class Instructors

National portal of India (https://www.india.gov.in/)



» Raksha Mantri Shri Rajnath Singh holds bilateral meetings with...

» Defence Secretary conducts bilateral meetings ahead of DefExpo...

» Department of Women, Children, Disabled and Senior Citizens...

» Indira Gandhi National Open University (IGNOU)

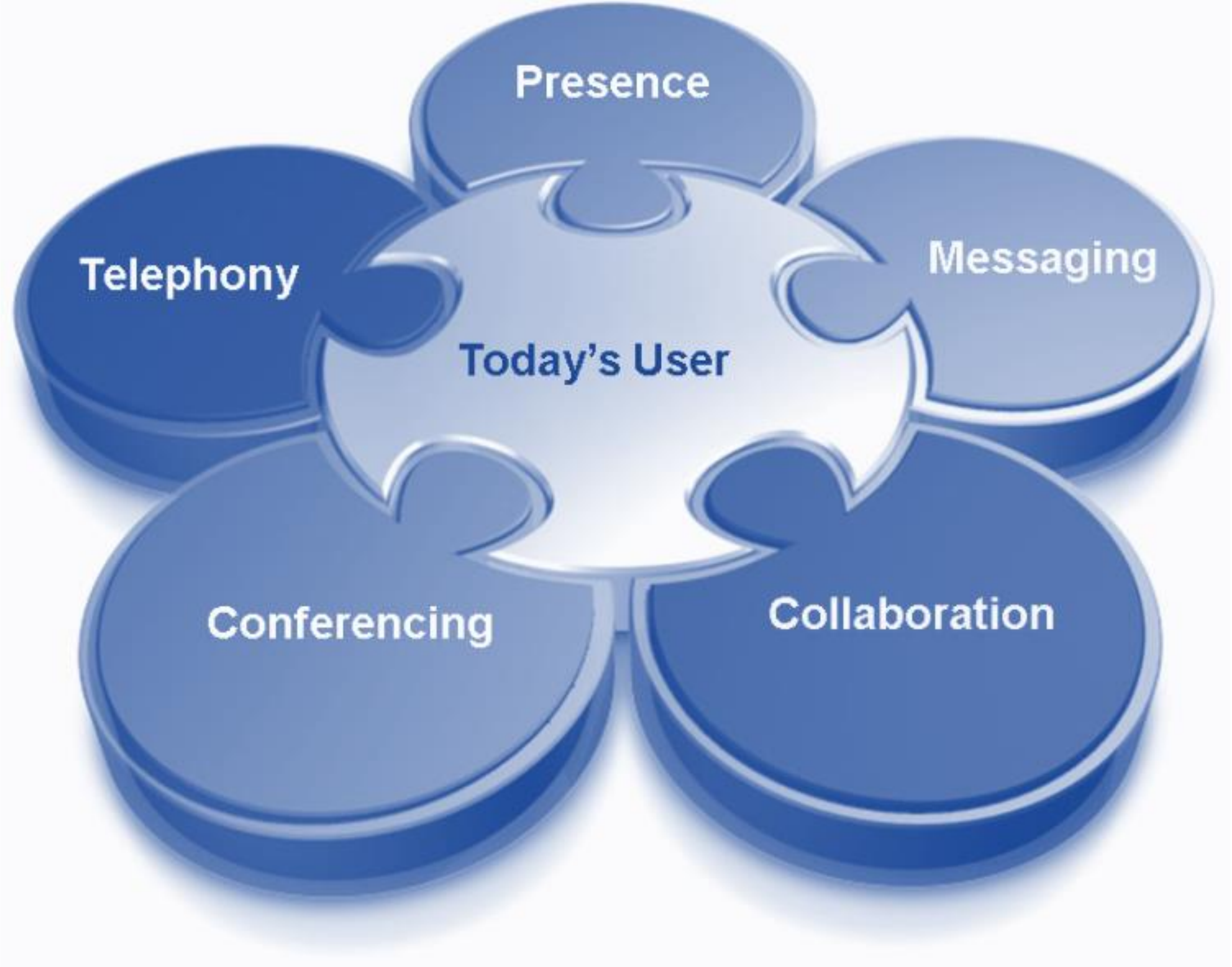
» Information on senior citizens schemes in Karnataka

initiative of the Government of

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	https://static.vikaspedia.in/media/files_en/education/Digital%20Litercy/Email5.pdf https://www.nidirect.gov.uk/articles/email-internet-and-social-media
		7	Collaborative Digital platforms	https://en.wikipedia.org/wiki/Digital_collaboration
		8-9	Tools/platforms for online learning, Collaboration using file sharing, messaging, video conferencing	https://vikaspedia.in/education/interactive-resources/digital-learning-resources https://www.prodigygame.com/in-en/blog/virtual-learning-tools/ https://dexteredward.com/secure-collaboration/

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
  - 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	https://safety.google/security-privacy/
		11	Threats in the digital world: Data breach and Cyber Attacks	https://www.csoonline.com/article/3541148/the-biggest-data-breaches-in-india.html https://www.secureworld.io/industry-news/top-10-data-breaches-of-all-time https://onlinedegrees.und.edu/blog/types-of-cyber-security-threats/
		12	Blockchain Technology	https://blockchain.gov.in/ https://blockchain.gov.in/assets/docs/blockchain_informatics.pdf https://blockchain.gov.in/assets/docs/Whitepaper_30jan.pdf
		13	Security Initiatives by the Govt of India	https://safety.google/ https://www.aicte-india.org/CyberSecurity https://www.niti.gov.in/sites/default/files/2019-07/CyberSecurityConclaveAtVigyanBhavanDelhi_1.pdf

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

- 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

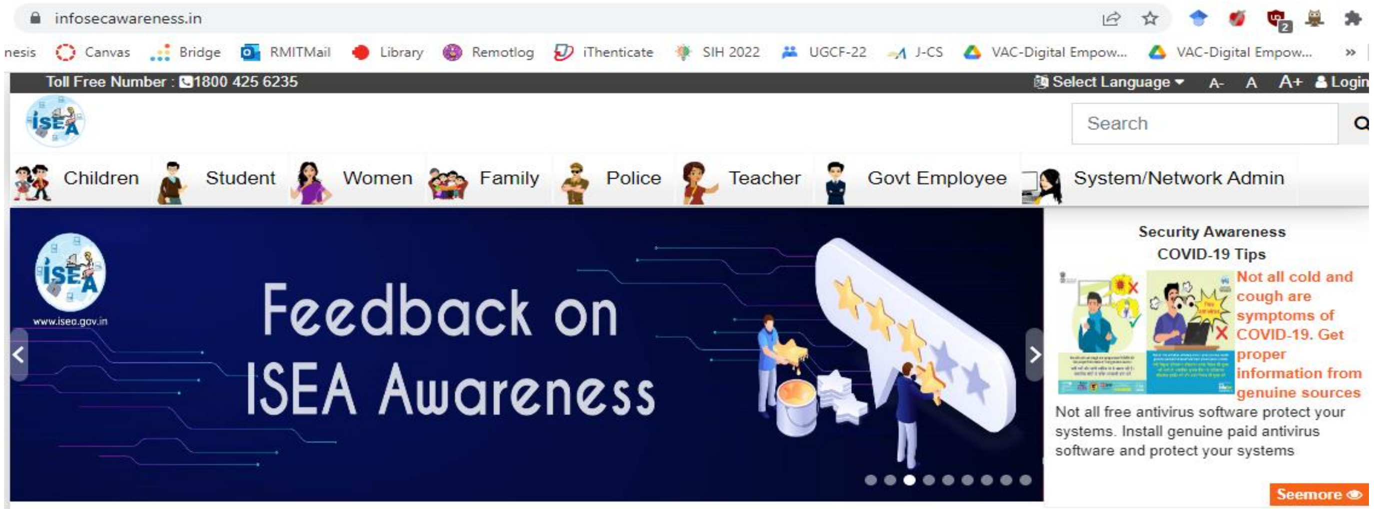
- cert-in.org.in

certin
Enhancing Cyber Security in India

- https://www.csk.gov.in/

साइबर स्वच्छता केन्द्र
CYBER SWACHHTA KENDRA
Botnet Cleaning and Malware Analysis Centre

- https://infosecawareness.in/


infosecawareness.in

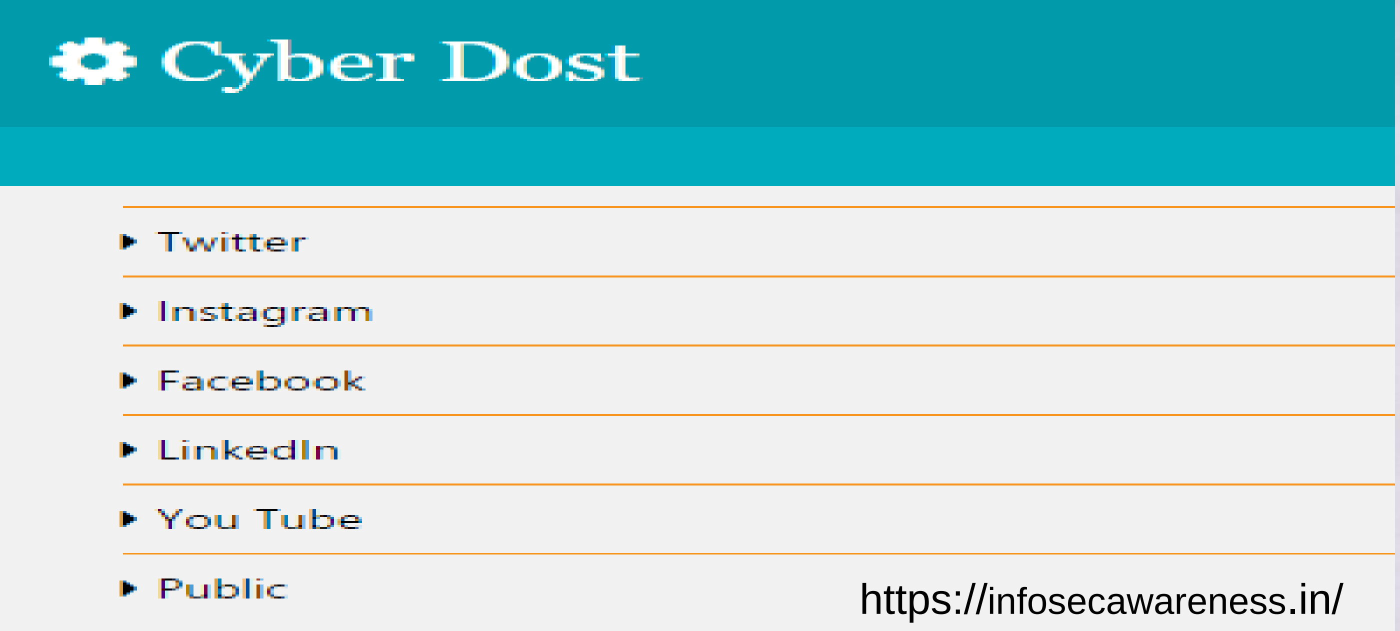
Toll Free Number : 1800 425 6235

Feedback on ISEA Awareness

Security Awareness
COVID-19 Tips

Not all cold and cough are symptoms of COVID-19. Get proper information from genuine sources

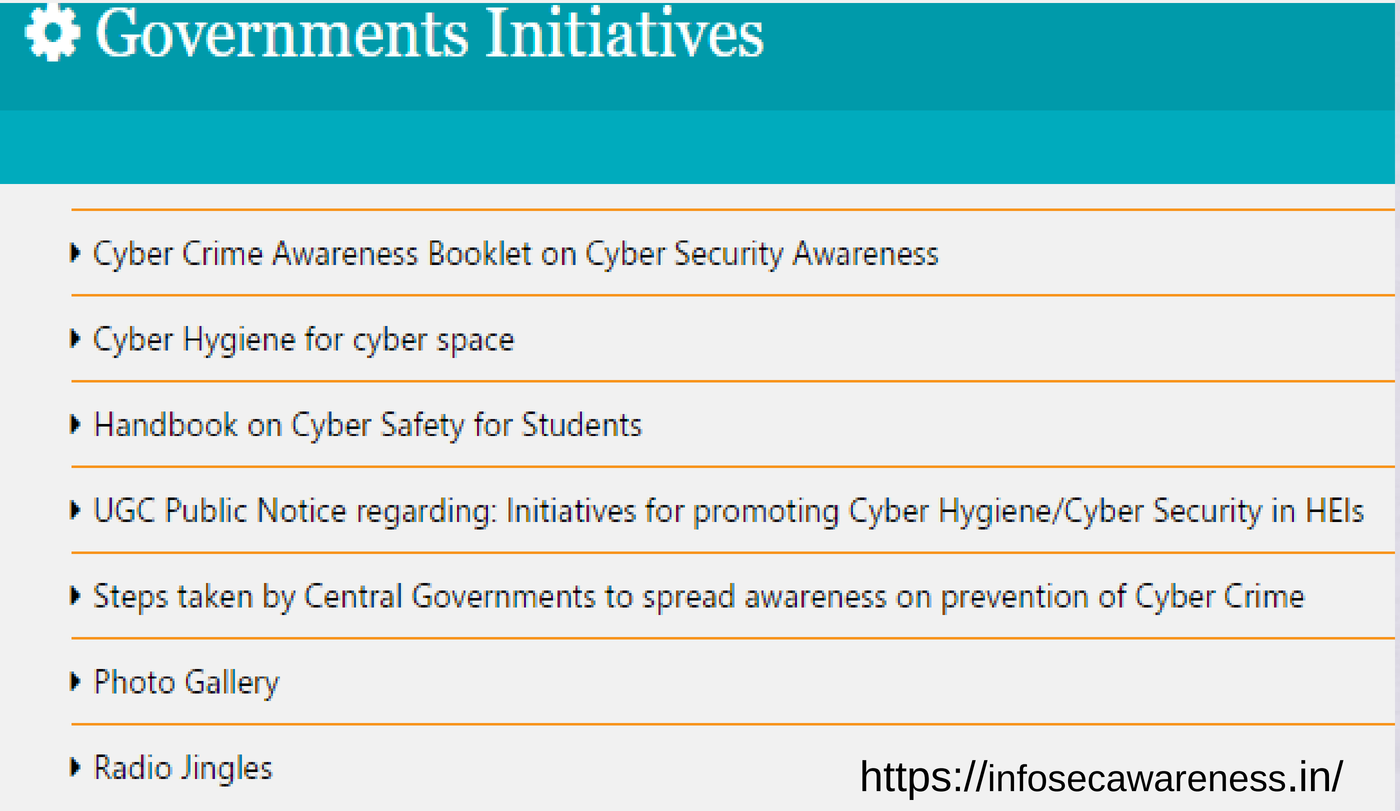
Not all free antivirus software protect your systems. Install genuine paid antivirus software and protect your systems



Cyber Dost

- Twitter
- Instagram
- Facebook
- LinkedIn
- You Tube
- Public

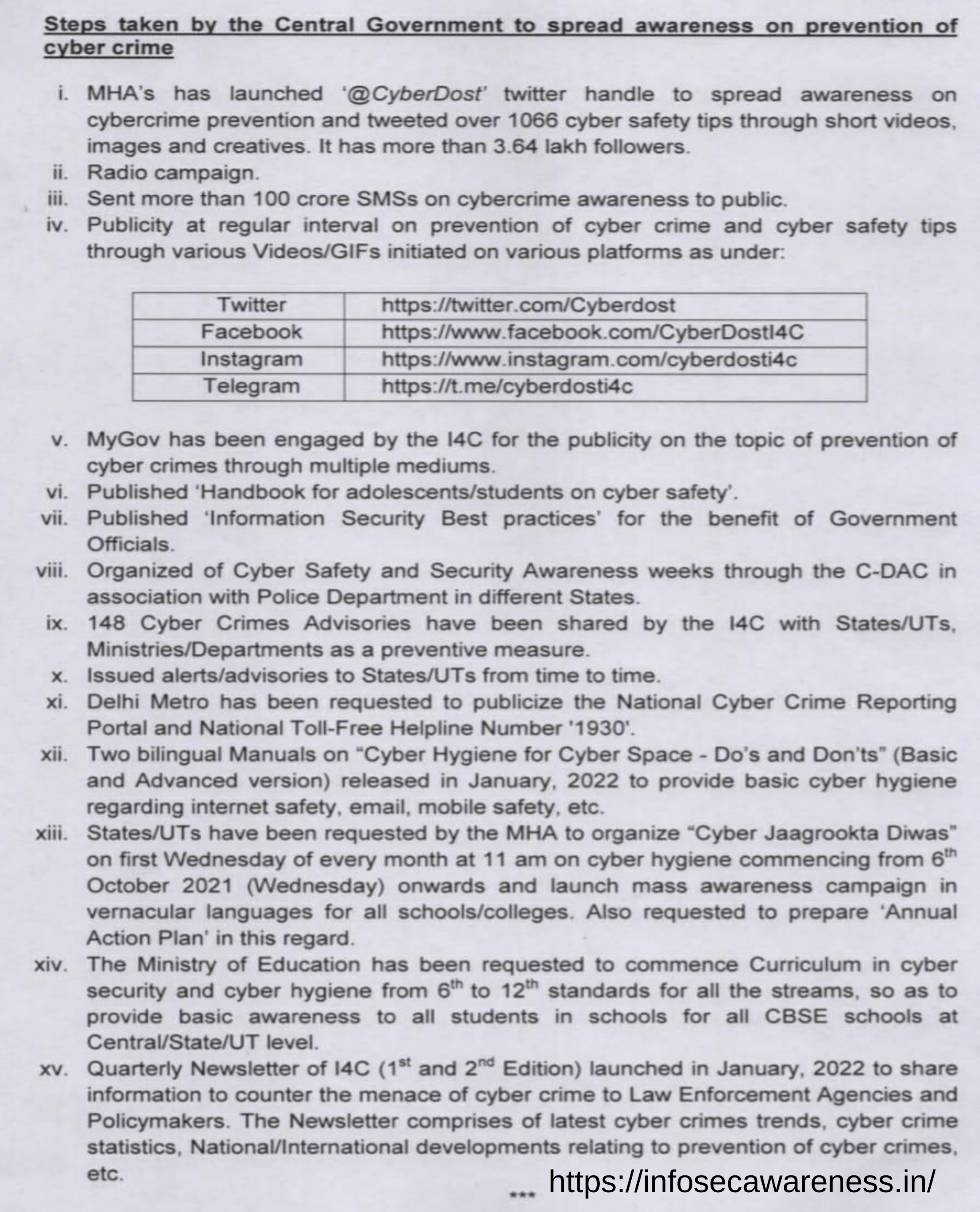
https://infosecawareness.in/



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/



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	https://twitter.com/Cyberdost
Facebook	https://www.facebook.com/CyberDostI4C
Instagram	https://www.instagram.com/cyberdosti4c
Telegram	https://t.me/cyberdosti4c

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 6th 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 6th to 12th 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 (1st and 2nd 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/

Unit-4: Ethical Issues in Digital World

Unit	Unit name	Week No	Topic	Teaching Resources
IV	Ethical Issues in Digital World	14	Netiquettes	https://www.kaspersky.com/resource-center/preemptive-safety/what-is-netiquette https://www.britannica.com/topic/netiquette
		15	Ethics in digital communication, Ethics in Cyberspace	https://onlinelibrary.wiley.com/doi/book/10.1002/9781118968888 https://iccopr.com/wp-content/uploads/2017/10/Ethics-in-Digital-Communications-Guidelines.pdf https://link.springer.com/chapter/10.1007/978-90-481-2370-4_1

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”. 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?