

**Department of Computer Science**

**University of Delhi**

**Curriculum**

**Multidisciplinary Courses of Study with Two Core Disciplines**

**under UGCF 2022**

**University of Delhi**

**Approved in**

**UG Committee meeting held on May 17, 2022**

**Faculty of Mathematical Sciences meeting held on May 25, 2022**

1. [Table of DSCs](#)
2. [Table of Discipline Courses](#)

**Table of DSCs**

Semester	DSC -No.	Title	L	T	P	Total credits	Desirable but not Prerequisite
I	<a href="#">DSC 01</a>	Introduction to Programming using C++	3	0	1	4	Nil
II	<a href="#">DSC 02</a>	Data structures	3	0	1	4	<a href="#">DSC 01</a>
III	<a href="#">DSC 03</a>	Computer System Architecture	3	0	1	4	Nil
IV	<a href="#">DSC 04</a>	Operating systems	3	0	1	4	<a href="#">DSC01</a> , <a href="#">DSC 02</a>
V	<a href="#">DSC 05</a>	Database Management System II	3	0	1	4	Nil
VI	<a href="#">DSC 06</a>	Computer Networks	3	0	1	4	<a href="#">DSC 02</a> <a href="#">DSC 01</a> /a course in C/C++ at plus 2 level/** <a href="#">DSC 04</a>
VII	<a href="#">DSC 07</a>	Design and Analysis of Algorithms	3	1	0	4	<a href="#">DSC 02</a>
VIII	<a href="#">DSC 08</a>	Information Security	3	0	1	4	

**Table of Discipline Courses**

Semester	Discipline	Title	L	T	P	Total credits	Desirable but not Prerequisite
I	<a href="#">A1</a>	Programming Fundamentals Using Python	3	0	1	4	Nil
II	<a href="#">A2</a>	Data Visualization using Python	2	0	2	4	<a href="#">A1</a>
III	<a href="#">A3</a>	Database Management System I	3	0	1	4	Nil
IV	<a href="#">A4</a>	Introduction to Web Programming	3	0	1	4	Nil
V	<a href="#">A5</a>	Multimedia and its Applications	2	0	2	4	Nil
VI	<a href="#">A6</a>	Introduction to Cyber Security (under preparation)	3	0	1	4	

**Note: Batch size for Practicals will be (8-10) and Tutorials will be (12-15).**

**DSC 01: Introduction to Programming Using C++**

**Course Objective**

The course introduces Object Oriented Programming Language C++ with the objective to use object oriented features to develop efficient programs.

## Course Learning Outcomes

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

1. Use the built-in data types of C++ programs and write and execute a simple program
2. Apply selection and iteration to solve programming problems.
3. Understand and implement arrays and user defined functions.
4. Comprehend the concepts of Object Oriented programming.

**Unit 1 Introduction to C++:** Need and characteristics of Object-Oriented Programming, Structure of a C++ Program (main () function, header files, output, input, comments), compile and execute a simple program.

**Unit 2 Data types and Expressions:** Keywords, built in data types, variables and constants, naming convention, Input-Output statements, operators and their precedence, expressions, typecasting, library functions.

**Unit 3 Control Constructs in C++:** Decision making using selection constructs, iteration using looping constructs.

**Unit 4 Arrays, Pointers and User defined functions:** Defining and initializing single and multi-dimensional arrays, user defined functions, passing arguments to functions, returning values from functions, inline functions, default arguments, introduction to pointers.

**Unit 5 Classes and Objects:** Need and implementation of abstraction creating classes, objects as function arguments, modifiers and access control, constructors and destructors.

## References

1. E. Balaguruswamy, *Object Oriented Programming with C++*, 7<sup>th</sup> edition, McGraw-Hill Education, 2017.
2. Robert Lafore, *Object Oriented Programming in C++*, 4<sup>th</sup> edition, SAMS Publishing, 2008.

## Additional References

- (i) D.S. Malik, *C++ Programming: From Problem Analysis to Program Design*, 6<sup>th</sup> edition, Cengage Learning, 2013.
- (ii) Herbert Schildt, *C++: The Complete Reference*, 4<sup>th</sup> Edition, McGraw Hill, 2003.

### Suggested Practical List

1. Write a program to find the largest of n natural numbers.
2. Write a program to find whether a given number is prime or not.
3. Write a program that takes a positive integer n and the produce n lines of output as shown:  
\*  
\* \*  
\* \* \*  
\* \* \* \*  
(for n = 4)
4. Write a menu driven program for following:
  - a. to check whether a given number is odd or even.
  - b. display a fibonacci series
  - c. compute factorial of a number
5. Write a program to accept a number, reverse it and print the sum of its digits.
6. Write a program using functions to print the series and its sum:  
 $1 + 1/2! + 1/3! + \dots + 1/n!$
7. Write a program to perform the following operations on an input string
  - a. Print length of the string
  - b. Find frequency of a character in the string
  - c. Print whether characters are in uppercase or lowercase
  - d. to check whether a given string is palindrome or not.
8. Write a program that will prompt the user for a list of 5 prices. Compute the average of the prices and find out all the prices that are higher than the calculated average.

9. Design a class named Vehicle, having registration number and year as its private members. Define a suitable constructor and a method to print the details of a vehicle. Write a C++ program to test the above class.
10. Inherit a class Car from the Vehicle class defined above. Add model to the Car class. Define a suitable constructor and a method to print the details of a car. Write a C++ program to test inheritance of this class.

## DSC 02: Data Structures

### Course Objective

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

### Course Learning Outcomes

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

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

### Syllabus

**Unit 1 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.

**Unit 2 Searching and Sorting :** Linear Search, Binary Search, Insertion Sort, Count Sort.

**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.

**Unit 5 Binary Search Trees:** insert, delete (by copying), search operations.

## References

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

## Additional references

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

## Suggested Practical List

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

## DSC 03: 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 Introduction:** Logic gates, Boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, memory units.

**Unit 2 Data Representation and Computer Arithmetic (of integers):** Number systems, complements, fixed and floating-point representation, character representation, Addition, subtraction, Magnitude comparison.

**Unit 3 Central Processing Unit:** Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control

**Unit 4 Basic Computer Organization and Design:** Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt.

**Unit 5 Programming the Basic Computer:** Instruction formats, addressing modes, instruction codes, machine language, assembly language.

**Unit 6 Input – Output Organization:** Peripheral devices, I/O interface, asynchronous data transfer, priority interrupt.



## References

1. M. Morris Mano, *Computer System Architecture*, 3<sup>rd</sup> edition, Pearson Education, 2017.

## Additional References

- (i) D. Comer, *Essentials of Computer Architecture*, 2<sup>nd</sup> edition, CRC Press, 2017.
- (ii) Linda Null, Julia Lobur, *Essentials of Computer Organization and Architecture*, 5<sup>th</sup> Edition, 2019.

## Suggested Practical List

1. Write a program to convert unsigned number from binary to octal, hex to decimal.
2. Write a program to convert unsigned number from decimal to binary, octal, hex.
3. Write a program that will prompt for the input of two integer values. Then using the bitwise shift operators show the result of
  - a) Left shifting the first number by the second.
  - b) Right shifting the first number by the second.
4. Write a program that will prompt for the input of two integer values. Then using the bitwise shift operators show the result of
  - a) Exclusive OR of the first number by the second bitwise.
  - b) OR of the first number by the second bitwise.
  - c) AND of the first number by the second bitwise.
5. Write a program that will prompt for the input of a binary value and display its 1's complement value.
6. Write a program that will prompt for the input of a binary value and display its 2's complement value.
7. Write a program that will prompt for the two inputs to implement binary addition operation.
8. Write a program that will prompt for the two input to implement binary subtraction using 2's complement.

## DSC 04: 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 system and their design.

### **Course Learning Outcomes**

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

### **Syllabus**

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

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

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

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

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

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

### **References**

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

### **Additional References**

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

### **Suggested Practical List**

1. Usage of following commands: ls, pwd, cat, who, rm, mkdir, rmdir, cd.
2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, bc.
4. Write a shell script to display date in the mm/dd/yy format.
5. Write a shell script to display the multiplication table of a given 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 a 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 has 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.

## **DSC 05: Database Management System II**

### **Course Objective**

The course will give an overview of categories of data models used by database management systems and writing queries in relational algebra. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice to write complex queries, nested queries using aggregate functions in SQL and to use basic database administration commands.

### **Course Learning Outcomes**

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

1. Write queries using relational algebra, a procedural language.
2. Use the concept of functional dependencies to remove data anomalies and arrive at normalized database design.
3. Understand the data storage system, file organization and need of file indexing.
4. Learn the importance of transaction processing and concurrency control.
5. Write complex and nested SQL queries, and learn basic database administration commands.
6. Acquire information about emerging technologies.

**Unit 1 Data models and Relational Algebra:** Categories of data models, types of database users, Relational Algebra Operations from SET Theory, SELECT, PROJECT, JOIN, DIVISION Operations.

**Unit 2 Normalization:** Functional dependencies, minimal cover, normalizing database with multiple keys till 3NF, Boyce-Codd Normal Form

**Unit 3 Database Storage and index Structures:** Storage of data, file structure, file organization and its types: Sequential, Heap and Indexed, Hash file, need for file Indexing, single- and multi-level indexing.

**Unit 4 Transaction Processing:** Concurrent execution of transactions and their handling, ACID properties, need of data recovery and log files.

**Unit 5 Advanced SQL:** Nested and complex queries using Inner JOIN, Left JOIN, Right JOIN, Full JOIN, views, Database Administration Commands: COMMIT, ROLLBACK, drop database, control permissions etc.

**Unit 6 Emerging Technology:** Distributed Database Concepts, Introduction to emerging technologies like Data Warehousing and OLAP, Data Mining.

## References

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

## Additional References

- (i) Silberschatz, A., Korth, H.F., & Sudarshan, S. *Database System Concepts*, 8<sup>th</sup> Edition, McGraw Hill, 2019.
- (ii) Ramakrishnan, R. & Gehrke, J. *Database Management Systems*, 3rd Edition, Tata McGraw Hill Education, 2014.

## Suggested Practical List

MySQL might be used for practical purposes.

The following concepts must be introduced to the students:

- Multiple table queries (join on different and same tables)
- Nested select statements
- Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists, union, intersect, minus, etc.)
- Categorization using group by.....having
- Arranging using order by

## List of sample queries to be done using the above schema.

Create tables with relevant column and integrity constraints and populate the tables with data.

Perform the following queries on the database:

1. Display all the details of all employees working in the company.
2. Display ssn, lname, fname, address of employees who work in department no 7.

3. Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong' and who is a manager
4. Retrieve the name and salary of every employee working in department 'Production'
5. Display maximum and minimum salaries of each department
6. Retrieve all employee names whose address is in 'Bellaire' and working on at least one project
7. Modify the datatype of hours column to float in works\_on table.
8. Retrieve all employees who age is more than 35 years
9. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)
10. Add another column named "Join\_Date" in the employee table and "Address" in the department table
11. Retrieve the names of all employees who do not have supervisors
12. Retrieve SSN and department name for all employees
13. Retrieve the name and address of all employees who work for the 'Research' department
14. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate
15. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor
16. Retrieve all combinations of Employee Name and Department Name
17. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project
18. Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees
19. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name
20. Select the names of employees whose salary does not match with salary of any employee in department 10
21. Drop the column "Join\_Date" from the employee table
22. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee
23. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
24. Change the name of existing table DEPT\_LOCATIONS to DLOCATIONS
25. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary for each department. Display with proper headings
26. Find the sum of the salaries and number of employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department

27. Select the names of employees whose salary is greater than the average salary of all employees in department 10
  28. For each department, retrieve the department number, the number of employees in the department and their average salary
  29. For each project, retrieve the project number, the project name, and the number of employees who work on that project for more than 10 hours
  30. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively
  31. For each department having more than 10 employees, retrieve the department name, number of employees drawing more than 40,000 as salary
  32. Display employee names having no dependent in descending order along with their age
  33. For each department, find the number of female and number of male employees along with name of manager of that department
  34. Find the name and age of youngest employee in each department
  35. Change the name of MINIT attribute of an employee table to MNAME using alter command
  36. Create a view to keep names of employees and their department names if they are not working on any project
  37. Create a view to keep track of all employee names who are working on same projects as that of employee 'Franklin T. Wong'
  38. Execute system administrative commands like commit, rollback, granting control permissions, etc.
- 

## **DSC 06: 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 Review of:** 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*, 5<sup>th</sup> edition. Pearson Education, 2012
2. Forouzan, B. A. *Data Communication and Networking*, 4<sup>th</sup> edition. McGraw-Hill Education, 2017



## **Additional References**

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

## **Suggested Practical List**

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

## **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*, 4<sup>th</sup> edition, Prentice Hall of India, 2022.
2. Kleinberg, J., Tardos, E., *Algorithm Design*. 1<sup>st</sup> edition. Pearson, 2013.

## Additional References

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

## Suggested Practical List

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

## 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.

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

### Additional References

- (i) Pfleeger C.P., Pfleeger S.L., Margulies J. *Security in Computing*, 5<sup>th</sup> edition, Prentice Hall, 2015.
- (ii) Lin S., Costello D.J., *Error Control Coding: Fundamentals and applications*, 2<sup>nd</sup> edition, Pearson Education, 2004.
- (iii) Stallings W. *Cryptography and network security*, 7<sup>th</sup> edition, Pearson Education, 2018.
- (iv) Berlekamp E. *Algebraic Coding Theory*, World Scientific Publishing Co., 2015.
- (v) Stallings W. *Network security essentials Applications and Standards*, 6<sup>th</sup> edition, Pearson Education, 2018.
- (vi) Whitman M.E., Mattord H.J., *Principle of Information Security*, 6<sup>th</sup> edition, Cengage Learning, 2017.
- (vii) Bishop M., *Computer Security: Art and Science*, 2<sup>nd</sup> Revised edition, Pearson Education, 2019.
- (viii) Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2<sup>nd</sup> edition, John Wiley & Sons, 2008.

### Suggested Practical List

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

9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.



## Course Objective

This course is designed as the first course that introduces programming concepts using Python to the students. The course focuses on the development of Python programming to solve problems of different domains

## Course Learning Outcomes

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

1. Understand the basics of programming language
2. Develop and document modular Python programs.
3. Apply suitable programming constructs and built-in data structures to solve a problem.

## Syllabus

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

**Unit 2 Creating Python Programs:** Identifiers and keywords; literals, numbers, and strings; Operators and expressions; Input and output statements; control structures (conditional statements, loop control statements, break, continue and pass), Errors and exception handling.

**Unit 3 User Defined Functions:** Defining functions, passing arguments and returning values, default arguments

**Unit 4 Built-in data structures:** Strings, Lists, Tuples, Sets, Dictionaries; their built-in functions, operators and operations.

## References

1. Kamthane, A. N., & Kamthane, A.A. *Programming and Problem Solving with Python*, McGraw Hill Education. 2017.
2. Balaguruswamy E., “*Introduction to Computing and Problem Solving using Python*”, 2<sup>nd</sup> Edition, McGraw Hill Education, 2018.

3. Taneja, S., Kumar, N. *Python Programming- A modular Approach*. Pearson Education India, 2018.

### Additional References

- (i) Guttag, J. V. *Introduction to computation and programming using Python*. MIT Press. 2018
- (ii) Downey, A. B. *Think Python–How to think like a Computer Scientist* 2nd Edition. O'Reilly 2015

### Suggested Practical List

1. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:

Grade A : if Percentage  $\geq 80$   
Grade B : if Percentage  $\geq 60$  and Percentage  $< 80$   
Grade C : if Percentage  $\geq 40$  and Percentage  $< 60$   
Grade D : if Percentage  $\leq 40$

2. WAP to print factors of a given number.
3. WAP to add N natural numbers and display their sum.

4. WAP to print the following conversion table (use looping constructs):

Height(in Feet)	Height(in inches)
5.0ft	60 inches
5.1ft	61.2inches

5.8ft	69.6inches
5.9ft	70.8inches

6.0ft	72inches
-------	----------

5. WAP that takes a positive integer n and the produce n lines of output as shown:

```
*
* *
* * *
* * * *
(for n =4)
```

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.
7. Write a function that calculates factorial of a number n.
8. WAP to print the series and its sum: (use functions)
- $$1/1! + 1/2! + 1/3! + \dots + 1/n!$$
9. WAP to perform the following operations on an input string
- Print length of the string
  - Find frequency of a character in the string
  - Print whether characters are in uppercase or lowercase
10. WAP to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.
11. WAP to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.
12. WAP to create a tuple t1 = (1,2,5,7,2,4). The program should perform the following:
- Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.



- b. Concatenate tuple  $t2 = (10,11)$  with  $t1$ .

## **A2 : Data Visualization using Python**

### **Course Objective**

This course introduces students to data visualization in the field of exploratory data science using Python.

### **Course Learning Outcomes**

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

1. Import and export data using files
2. Perform data handling using Numpy arrays.
3. Load data in Pandas data frames and its manipulation.
4. Visualize data using Matplot library.

### **Syllabus**

**Unit 1 Introduction :** Motivation for using Python for Data Visualization, Essential Python Libraries: NumPy, Pandas, Matplotlib, Import and Export of Data, Import and Export of data using files.

**Unit 2 Array manipulation using Numpy:** Numpy array: Creating Numpy arrays , Data Types for Numpy arrays, Arithmetic with NumPy Arrays Basic Indexing and Slicing, swapping axes, transposing arrays.

**Unit 3 Data Manipulation using Pandas:** Data Structures in Pandas: Series, DataFrame, Index objects, Loading data into Pandas data frame. Working with Data frames. Grouped and aggregate calculations

**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, boxplot.

### **References**

1. McKinney W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython*. 2nd edition. O'Reilly Media, 2018.

2. Chen D. Y, *Pandas for Everyone: Python Data Analysis*, Pearson, 2018.

### **Additional References**

(i) Molin S. *Hands-On Data Analysis with Pandas*, Packt Publishing, 2019.

### **Suggested Practical List**

Use data set of your choice from Open Data Portal ([https:// data.gov.in/](https://data.gov.in/)) for the following exercises.

1. Practicals based on NumPy
2. Practicals based on Pandas
3. Practicals based on Data Visualization using matplotlib.

## **A3 : Database Management System I**

### **Course Objective**

The course introduces the students to the fundamentals of database management systems. Students will learn about the importance of database structure and its designing using conceptual approach using Entity Relationship Model and formal approach using Normalization. They will learn SQL to construct the database, manipulate and retrieve its contents.

### **Course Learning Outcomes**

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

1. Understand the importance of a database system and its difference from the traditional file system processing.
2. Create conceptual data models using entity relationship diagrams for modeling real-life situations and map it to corresponding relational database schema.
3. Construct relational database and formulate queries for data retrieval and data update using SQL
4. Normalize a database upto 3NF to make it free from update anomalies.

**Unit 1 Database:** Introduction to database and DBMS, architecture of DBMS, data independence, components of database systems, difference between traditional file system and database management system.

**Unit 2 Entity-Relationship Modeling:** Entity types, types of attributes and key, relationships types, constraints on relationship, modeling ER diagram, Database design using ER diagrams.

**Unit 3 Relational Data Model:** Relational model concepts, relation characteristics, relational integrity constraints, types of keys, update anomalies.

**Unit 4 Normalization:** Functional dependencies, Use of normalization, First, Second and Third normal forms using single key.

**Unit 5 Introduction to Structured Query Language:** Overview of SQL query language, Data definition and manipulation languages, set operations. DDL: to construct the database and alter its contents, DML: Querying in SQL to retrieve data from the database, use of join and simple aggregate functions.

### **Suggested Practical List**

MySQL might be used for practical purposes.

### **Suggestive list of database schemes and corresponding queries:**

1. Create a database having the following three tables to store the details of students of the Computer Department in a college.

**Student (Rollno, Name, DOB, Address, Phoneno)**

**Paper Details (Paper\_code, Paper\_Name)**

**Academic details (Rollno, Paper\_code, Attendance, Marks)**

1. Identify primary and foreign keys. Create the tables with keys and insert at least 5 records in each table.
2. Design a query that will return the records (from the second table) along with the name of the student from the first table, related to students who have attendance more than 75 and marks more than 60 in paper P2.
3. List all students who live in “Delhi” and have marks greater than 60 in paper with code P1.
4. Find the total attendance and total marks obtained by each student.
5. List the names of the students who have got the highest marks in paper P2.
6. Find the average marks obtained by each student.

7. Update the name of the paper with Paper code P1 from “Introduction to Computers” to “Computer Science Fundamentals”.
8. List the names of the students who have got the lowest marks in paper “Computer Science Fundamentals”.
9. List the names of all the students whose name ends with ‘sh’.
10. List the total number of students who have secured above 90 in paper P4.
11. Add another column ‘Email’ in the Student table.
12. Update the phone number of the student with Roll number 34 to ‘9888812345’.
13. List all student details whose marks in paper with code P4 are between 90 and 95.
14. Display the names of all the students in increasing order of their marks in paper P3.
15. Display the names of all the students whose marks are greater than the marks of ‘Ramesh’ in paper P2.

2. Create the following tables with primary key and foreign keys, enter at least 5 records in each table and answer the queries given below.

**Suppliers (SNo, Sname, Status, SCity)**

**Parts (PNo, Pname, Colour, Weight, City)**

**Project (JNo, Jname, Jcity)**

**Shipment (Sno, Pno, Jno, Quantity)**

1. Get supplier numbers for suppliers in Guwahati with status greater than 20.
2. Get supplier details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.
3. Get suppliers names for suppliers who do not supply part P2.
4. For each shipment get full shipment details, including total shipment weights computed as  $\text{Weight} \times \text{Quantity}$  of corresponding parts.
5. Get all the shipments where the quantity is in the range 300 to 750 inclusive.
6. Get part nos. for parts that either weigh more than 1Kg or are supplied by suppliers S2 or both.
7. Get the names of cities that store more than two red parts.
8. Update the city of S1 supplier to “Delhi”.
9. Get part numbers for parts supplied by a supplier in Allahabad to a project in Chennai.
10. Find the names of all parts whose color starts with the letter b.
11. Change the datatype of the weight attribute in the Parts table from int to float.
12. Find the number of parts of each color.

13. Find the names of all the projects which are located in the city Jaipur and in which the part is supplied by supplier S3.
14. Delete all the projects which are located in Madras.

## References

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

## Additional References

- (i) Silberschatz, A., Korth, H.F., & Sudarshan, S. *Database System Concepts*, 8<sup>th</sup> Edition, McGraw Hill, 2019.
- (ii) Ramakrishnan, R. & Gehrke, J. *Database Management Systems*, 3<sup>rd</sup> Edition, Tata McGraw Hill Education, 2014.

## A4: 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, Javascript and jQuery.
3. Learn to validate client-side data.

### Syllabus

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

architecture.

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

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

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

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

## References

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

## Additional References

- (i) Minnick, J. (2015). Web Design with HTML5 and CSS3. 8th edition. Cengage Learning.
- (ii) Boehm, A., & Ruvalcaba, Z. (2018). Munarch's HTML5 and CCS(4th Edition). Mike Murach & Associates.
- (iii) J. A. Ramalho (2007), Learn Advanced HTML 4.0 with DHTML, BPB Publications

- (iv) Ivan Bayross (2009), Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI , BPB Publications.

## **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 leftframe.
1. Create a student registration form using HTML which has the following controls:
    - I. Text Box
    - II. Dropdown box
    - III. Option/radio buttons
    - IV. Check boxes
    - V. Reset and Submit button

### **CSS**

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

### **Javacript**

Create event driven programs 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.

Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customise the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:

- a. Roll number is a 7-digit numeric value
- b. Name should be an alphabetical value(String)
- c. Non-empty and valid fields like DOB

### **jQuery and JSON**

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

## **A5 : Multimedia and its Applications**

### **Course Objective**

To learn about creativity and imagination for making them self- motivated artists with strength in creative vision and concept development.

### **Course Learning Outcomes**

Introduce multimedia with the fundamentals of Art and Design, which will encourage the students for creativity and imagination. It includes Graphic Design for creating effective visual communication and web Design for global communication through a website and it offers very exciting opportunities for good jobs to students.



## Syllabus

**Unit 1 Introduction:** Definition Multimedia and uses, applications, Hardware and Software requirements, multimedia presentation and authoring tools, LAN and multimedia, multimedia servers and database: vector graphics, 3-D graphics program, animation techniques, shading, anti-aliasing, morphing: video on demand.

**Unit 2 Text and Graphics:** Uses of text in multimedia, families and faces of fonts, outline fonts, bitmap fonts, International characters sets and hypertext digital font techniques.

Graphics- Introduction, types, color and color models, color palettes, vector drawing, 3-D drawing and rendering, making still images, editing and capturing images.

**Unit 3 Audio and Video:** Digital representation of sound, MIDI audio, MIDI vs. Digital Audio, Audio file formats, Adding sound to your multimedia project, Audio software and hardware, Video basic, how video works, broadcast video standard, Analog video, Digital video, shooting and editing video, video compression and file format.

**Unit 4 Animation and multimedia authoring:** Introduction, principles, types and uses, cell animation, computer animations, morphing, Animation software compression techniques: need and types of compression- lossy and lossless, CODECs, authoring basic, types of authoring tools.

**Unit 5 Multimedia on the web:** Introduction, Bandwidth, text on the web- Dynamic and embedded font technology, Audio and video on the web, buffering and streaming, webcasting, video conferencing.

## References

1. Multimedia: Making It Work by Tay Vaughan, TataMcGraw-Hill
2. Computer Graphics and Multimedia by A. P. Godse, Technical Publication
3. Multimedia: An Introduction, Villamil and Molina, PHI
4. Sound and Video, Lozano. Multimedia, PHI

## Suggested Practical List

1. Create an animated scene using the tools panel and the properties panel.
2. Create an animation using Free transform tool that should use

- Move objects
  - Skew objects
  - Rotate objects
  - Stretch objects while maintaining proportion
3. Create an animation of Text using font, size, color and lighting effect.
  4. Modify the document (changing background color etc.) using the following tools
    - Eraser tool
    - Hand tool
    - Ink bottle tool
    - Zoom tool
    - Paint Bucket tool
    - Eyedropper tool
  5. Create an animated Flag
  6. Create a scene to show animated walk and run cycles.(using multiple layers and motion tweening)
  7. Create an animation having an images having morphing effect.
  8. Create an animation using (Shape Tweening and shape hints) for transforming one shape into another.
  9. Create an animation of a cartoon picture using sound.
  10. Create an animation of a character using key frames.
  11. Create an animation for a bouncing ball.(use motion guide layer)
  12. Create an animated scene of rain using the ripple effect.

## **A6 : Introduction to Cyber Security**

Under Preparation