

# Semester II

## DISCIPLINE SPECIFIC CORE COURSE -4 (DSC-4): Engineering through Linear Algebra

### CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Engineering through Linear Algebra, DSC-4	4	3	0	1	Class XII pass	Mathematics till XII

#### Learning Objectives

Almost every area of modern science contains models where equations may be approximated by linear equations and linear algebra plays a vital role for finding their solution and interpreting them. This paper aims to enable the student to learn linear models for various physical problems such as traffic flow, electric-circuit flow etc. and to facilitate their solution using concepts of linear dependence, independence, rank, basis, eigenvalues, eigenvectors etc. This paper intends to provide geometrical interpretation of vectors, basis and vector operations in 2 & 3 dimensions and lays the groundwork for a more abstract, pure-mathematical treatment of vector spaces. Also, the importance and application of eigenvalues, eigenvectors in computer graphics, face recognition and many other fields is taught. Students will also learn how to use MATLAB for some simple matrix operations, for finding eigenvalues & eigenvectors, rank etc.

#### Learning outcomes

After completing this course, student should be able to;

- Understand graphical representation of vector and their operations in 2 and 3 dimensions
- Solve linear matrix system  $AX=B$
- Understand the concept of Eigen values and Eigen vectors and their applications in computer graphics, face recognition algorithms & many other fields
- Conceptualize vector spaces, subspaces and their basis functions

- Understand inner product spaces, orthogonal sets, projection and orthogonal diagonalisation
- Learn basic arithmetic operations of matrices in MATLAB
- Implement basic loops (for, while, if else etc) of programming in MATLAB
- Write their own programs for solving system of linear equations

## **SYLLABUS OF DSC-4**

### Unit I: (16 Hour)

#### Algebra

Algebra of matrices – Review of Determinants - Hermitian, Skew-Hermitian and Unitary matrices - Vectors and vector operations in 2 and 3 dimensions - Solution and application of linear matrix system  $AX = B$

### Unit II: (16 Hour)

#### Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors, minimal polynomial, Cayley-Hamilton theorem and diagonalization

### Unit III: (16 Hour)

#### Abstract vector spaces, subspaces

Finite dimensional vector spaces - Linear independence and dependence of vectors, bases, dimension of vector spaces - Finite dimensional inner product spaces

### Unit IV: (16 Hour)

#### Orthogonality

Orthogonal sets and projections, Gram Schmidt process, orthogonal diagonalisation

## **Practical component –**

Engineering Kitchen Activity (matrix based numerical mathematics software) [Laboratory]

- Basic arithmetic operations, hierarchy of arithmetic operations
- Declaration and assignment of variables
- Introduction to elementary mathematical functions
- Dealing with matrices and arrays
- Basic programming with loops (for, while, switch), if else statements
- Programs for solving system of linear equations, Orthogonalization
- Creating 2D, 3D plots
- Innovation project

### Essential/recommended readings

1. *Linear Algebra and its Applications*, D. C. Lay, Addison Wesley, 2005.
2. *A Modern Introduction*, David Poole, *Linear Algebra*, Brooks Cole, 2011.

**Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.**

**DISCIPLINE SPECIFIC CORE COURSE – 5 (DSC-5): Data Structure and Design**

**CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE**

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Data Structure and Design, DSC-5	4	3	1	0	Class XII pass	Programming Languages

**Learning Objectives**

This course objective is to give an understanding of the real world data representation, organisation and structuring to the student while writing the programs and software. The course makes them familiar with the several types of data structures and their strengths and weaknesses, particularly in a real-world situation.

**Learning outcomes**

- Introduction to Data structure and their significance.
- Practical and theoretical understanding of Dynamic optimization
- Basics of Memory Hierarchy and implementation
- Understanding and implementation of Hashing, Networks and Graphs
- Understanding basics and practical aspects of Searching algorithms in the real world through implementation.
- Introduction and implementation of Heaps and Priority Queues and their comparison with other data structure

**SYLLABUS OF DSC-5**

Unit I: (16 Hour)

Program and data analysis

Introduction to Data structure, Basic concepts of Correctness, Efficiency and Application, Dynamic optimization Concept, Search Algorithms

Unit II: (16 Hour)

Data items arrangements and processing

Sorting Algorithms, Introduction to Linear Data Structures: Linked List, Stack and Queues

Unit III: (16 Hour)

Hierarchical arrangements and processing

Introduction to Hierarchical Data structure: Tree, Introduction to Heap, Priority Queues and Hashing

Unit IV: (16 Hour)

Network arrangements and analysis

Networks arrangements, Complex systems and real-world studies, Computational analysis

Practical component:

**Engineering Kitchen Activity [Laboratory]:**

- Implementation of Linked list in C/C++
- Implementation of Trees in C/C++
- Implementation of variant of Trees in C/C++
- Implementation of Heaps in C/C++
- Implementation of Hashing in C/C++
- Implementation of Priority Queues in C/C++
- Implementation of Graph and Network based approaches in C/C++
- Innovation Project

Essential/recommended readings

1. *Algorithms and Data Structures*, N. Wirth, Prentice-Hall of India, 2009
2. *Data Structures and Algorithms in C++*, A. Drozdek, Course Technology, 2013

**Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.**

### DISCIPLINE SPECIFIC CORE COURSE– 6 (DSC-6): Object Oriented Programming

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Object	4	3	0	1	Class XII	Programming Languages

<b>Oriented Programming, DSC-6</b>					<b>pass</b>	
------------------------------------	--	--	--	--	-------------	--

### **Learning Objectives**

The objective is to implement real-world entities like inheritance, hiding, polymorphism etc. in programming. To learn how to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

### **Learning outcomes**

Upon Completion of this course the students will be able to:

- Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
- Use NetBeans, Eclipse, BlueJ as an Integrated Development Environment. Test a program and, if necessary, find mistakes in the program and correct them.
- Take a problem and develop the structures to represent objects and the algorithms to perform operations.
- Name and apply some object-oriented design patterns and give examples of their use.
- Apply standards and principles to write truly readable code.
- Design a class that serves as a program module or package.
- Design applications with an event-driven graphical user interface using java applets.
- Design different android applications such as web apps for the real-world problems.

### **SYLLABUS OF DSC-6**

#### **Unit I: (16 Hour)**

Introduction to Java, Introduction to byte code, security and portability, Data Types, variables, operators, arrays, type conversion and casting, type promotion, Control statements, standard input-output, Designing Classes, constructors, methods. access specifiers - public, private, protected

#### **Unit II: (16 Hour) Classes and Objects**

Introduction, Class revisited, constant objects and constructor, static data members with constructors and destructors, constructor overloading, nested classes, objects as arguments, returning objects, constant parameters and member functions, static data and member functions

### Unit III: (16 Hour)

Inheritance, packages and interfaces and Exception Handling Math, String, polymorphism - function overloading, function overriding, abstract classes, Dynamic objects - Introduction, array of objects, Exception types, nested try-catch, throw, throws and finally statements

### Unit IV: (16 Hour)

Multi Thread Programming, Thread creation, synchronization and priorities

Practical component –

Engineering Kitchen Activities [Laboratory]

- Programs implying the use of Arrays, Linked Lists, Strings, Loops
- Programs on Object & Classes from Realistic Environment and Systems
- Programs demonstrating Constructors, Destructors, Methods & other concepts
- Programs Showcasing Inheritance, Polymorphism, Encapsulation & other OOPS features
- Programs on Exception Handling, Packages and Threading
- Reverse Engineering a Java Source/ project/Mobile Application and understanding the concepts
- Mapping the programs with Real life Systems and showcasing the implementation
- Innovation project

Essential/recommended readings

1. *Java: The Complete Reference*, 10<sup>th</sup> Edition. Herbert Schildt. McGraw-Hill, 2017.
2. *C++: The Complete Reference*, 4<sup>th</sup> Edition. Herbert Schildt. McGraw-Hill, 2012.
3. *Object Oriented Programming with C++*, 6<sup>th</sup> Edition. E Balagurusamy. Tata McGraw-Hill, 2001.
4. *C++ For Artists: The Art, Philosophy, and Science Of Object-Oriented Programming*. Rick Miller, Pulp Free Press, 2008
5. *Java For Artists: The Art, Philosophy, and Science Of Object-Oriented Programming*. Rick Miller , Pulp Free Press, 2008