

**DISCRETE MATHEMATICS & GRAPH THEORY**

<b>Course Code</b>	<b>23BS1305</b>	<b>Year</b>	<b>II</b>	<b>Semester</b>	<b>I</b>
<b>Course Category</b>	<b>BS&amp;H</b>	<b>Branch</b>	<b>CSE</b>	<b>Course Type</b>	<b>Theory</b>
<b>Credits</b>	<b>3</b>	<b>L-T-P</b>	<b>3-0-0</b>	<b>Prerequisites</b>	<b>Basic Mathematics</b>
<b>Continuous Internal Evaluation:</b>	<b>30</b>	<b>Semester End Evaluation:</b>	<b>70</b>	<b>Total Marks:</b>	<b>100</b>

<b>Course Outcomes</b>	
Upon successful completion of the course, the student will be able to	
<b>CO1</b>	Understand the fundamental concepts of discrete mathematics and graph theory. <b>(L2)</b>
<b>CO2</b>	Apply mathematical techniques to prove arguments / statements. <b>(L3)</b>
<b>CO3</b>	Apply various method(s) for solving different recurrence relations. <b>(L3)</b>
<b>CO4</b>	Analyze various graphs by their characteristics to construct a tree. <b>(L4)</b>

<b>SYLLABUS</b>		
<b>Unit No.</b>	<b>Contents</b>	<b>Mapped CO</b>
<b>I</b>	<b>Mathematical Logic:</b> Introduction-Statements and notations-Connectives (Negation, Conjunction, Disjunction)-Statement formulas and Truth tables, Conditional and Bi-conditional, Well-Formed Formulas, Tautologies, Equivalence of formulas, Duality law, Tautological Implication, Functionally Complete sets of Connectives, Other Connectives. ( NAND, NOR, XOR) <b>Normal Forms:</b> Disjunctive Normal Forms (DNF), Conjunctive Normal Forms (CNF), Principal of Disjunctive Normal Forms (PDNF), Principal of Conjunctive Normal Forms (PCNF).	<b>CO1,CO2</b>
<b>II</b>	<b>Theory of Inference for Statement Calculus:</b> Validity using truth tables-Rules of Inference – Consistency & Inconsistency of Premises and Indirect method proof. Predicate calculus: Introduction to Predicates - Statement functions, Variable and Quantifiers- Predicate formulas-Free and Bound Variables-Universe of Discourse.	<b>CO1,CO2</b>
<b>III</b>	<b>Recurrence Relations</b> -Generating functions of sequences – Recurrence relations- Solving recurrence relations by substitution, method of characteristic roots-Solution of Inhomogeneous Recurrence relations.	<b>CO1,CO3</b>
<b>IV</b>	<b>Relations and Directed Graphs</b> -Special Properties of Binary Relations- Equivalence Relations- Ordering Relations-Poset diagrams, Special elements in Posets-Lattices- Operations on Relations- Representation of relation. Graphs- Basic Concepts- Operations on Graph-Matrix representation of Graph-Adjacency Matrix, Incidence Matrix-Paths and Closures- Warshall's Algorithm-and Sub graphs –Isomorphic Graphs- Directed Graphs	<b>CO1,CO4</b>
<b>V</b>	<b>Planar Graphs</b> -Euler's Formula- Multi-graphs and Eulerian Graphs-Hamiltonian Graphs-Chromatic Number. Trees and Their Properties - Spanning Trees-Breadth First and Depth First Spanning Trees –BFS and DFS algorithms-Minimal Spanning tree-Prim's and Kruskal's algorithms.	<b>CO1,CO4</b>

**Learning  
Resources**

**Text Books:**

1. Discrete Mathematical Structures with Applications to Computer Science , J P Trembly and R Manohar, 1988, McGraw-Hill
2. Discrete Mathematics for Computer Scientists & Mathematicians, Joe L. Mott. Abraham Kandel and Theodore P. Baker, Second Edition, 2017, PHI.

**Reference Books**

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, Seventh Edition, 2017, McGraw-Hill.
2. Discrete Mathematics, Swapna Kumar Chakraborty, BikashKanti Sarkar, First Edition, 2011, Oxford University Press

**E-Resources**

1. <https://www.geeksforgeeks.org/engineering-mathematics-tutorials/>
2. [https://www.tutorialspoint.com/discrete\\_mathematics/index.htm](https://www.tutorialspoint.com/discrete_mathematics/index.htm)
3. <http://www.alas.matf.bg.ac.rs/~mi10164/Materijali/DS.pdf>
4. <https://nptel.ac.in/courses/111107058/>

**DIGITAL LOGIC & COMPUTER ORGANIZATION**

<b>Course Code</b>	<b>23ES1304</b>	<b>Year</b>	<b>II</b>	<b>Semester</b>	<b>I</b>
<b>Course Category</b>	<b>Engineering Science</b>	<b>Branch</b>	<b>CSE</b>	<b>Course Type</b>	<b>PC</b>
<b>Credits</b>	3	<b>L – T – P</b>	3-0-0	<b>Prerequisites</b>	Engineering Mathematics, BEEE
<b>Continuous Evaluation:</b>	30	<b>Semester End Evaluation:</b>	70	<b>Total Marks:</b>	100

<b>Course Outcomes</b>		
Upon successful completion of the course, the student will be able to:		
<b>CO1</b>	Understand the basic concepts of digital circuits, computer functional units of computer system and its organization, computer arithmetic.	L2
<b>CO2</b>	Understand the basic concepts of I/O organization and Processor Organization	L2
<b>CO3</b>	Apply the minimization techniques to simplify Boolean expressions	L3
<b>CO4</b>	Apply the functionality of combinational circuits and sequential circuits.	L3

<b>Syllabus</b>		
<b>Unit No.</b>	<b>CONTENTS</b>	<b>Mapped CO</b>
<b>I</b>	<b>Data Representation:</b> Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements of Numbers, Signed binary numbers, Binary codes, Basic Gates <b>Digital Logic Circuits-I:</b> Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Canonical and Standard forms, The Map Method, Four-Variable K-map, Product of Sums simplification, Don't Care Conditions	<b>CO1,CO3</b>
<b>II</b>	<b>Digital Logic Circuits-II:</b> Combinational Circuits, Analysis of Combinational circuits, Binary Adder – Subtractor, Decoders, Encoders, Multiplexers <b>Sequential Circuits</b> – Latches, Flip-Flops, Shift Registers, Ripple counters, Synchronous Counters	<b>CO1,CO4</b>
<b>III</b>	<b>Processor Organization:</b> General Register Organization, Stack Organization, Instruction Formats and Addressing Modes <b>Computer Arithmetic:</b> Addition and Subtraction, Multiplication Algorithms, Decimal Arithmetic Unit, Decimal Arithmetic Operations	<b>CO1,CO2</b>
<b>IV</b>	<b>The Memory Organization:</b> Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory and Virtual Memory	<b>CO1</b>
<b>V</b>	<b>Input/output Organization:</b> Peripheral Devices, Input Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA	<b>CO2</b>

<b>Learning Resources</b>	
<b>Text Books</b>	
<ol style="list-style-type: none"> <li>1. Digital Design, 6<sup>th</sup> Edition, M. Morris Mano, Pearson Education.</li> <li>2. Computer Systems Architecture, M.Moris Mano, Revised 3<sup>rd</sup> Edition, Pearson</li> <li>3. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 6<sup>th</sup> edition, McGraw Hill</li> </ol>	
<b>Reference Books</b>	
<ol style="list-style-type: none"> <li>1. Computer Organization and Design, David A. Paterson, John L.Hennessy, Elsevier</li> <li>2. Fundamentals of Logic Design, Roth, 5<sup>th</sup> Edition, Thomson</li> <li>3. Computer Organization and Architecture, William Stallings, 11<sup>th</sup> Edition, Pearson.</li> </ol>	
<b>E-Resources &amp; other digital material</b>	
<ol style="list-style-type: none"> <li>1. <a href="https://nptel.ac.in/courses/117105080">https://nptel.ac.in/courses/117105080</a></li> <li>2. <a href="https://archive.nptel.ac.in/courses/106/105/106105163/">https://archive.nptel.ac.in/courses/106/105/106105163/</a></li> <li>3. <a href="https://nptel.ac.in/courses/106/103/106103068/">https://nptel.ac.in/courses/106/103/106103068/</a></li> </ol>	

### ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS

<b>Course Code:</b>	<b>23CS3301</b>	<b>Year:</b>	<b>II</b>	<b>Semester:</b>	<b>I</b>
<b>Course Category:</b>	Professional Core Course	<b>Branch:</b>	CSE	<b>Course Type:</b>	Theory
<b>Credits:</b>	3	<b>L-T-P:</b>	3-0-0	<b>Prerequisites:</b>	Data Structures through C / Object Oriented Programming
<b>Continuous Internal Evaluation:</b>	30	<b>Semester End Evaluation:</b>	70	<b>Total Marks:</b>	100

#### COURSE OUTCOMES

Upon successful completion of the course, Student will be able to

<b>CO1</b>	Understand the fundamental concepts of algorithm analysis and design techniques.	<b>L2</b>
<b>CO2</b>	Apply various algorithm design techniques for solving problems	<b>L3</b>
<b>CO3</b>	Apply the concepts of Trees and Graphs for solving problems effectively.	<b>L3</b>
<b>CO4</b>	Analyze the given scenario and choose appropriate algorithm design for solving problems.	<b>L4</b>

#### SYLLABUS CONTENTS

<b>Unit No.</b>	<b>SYLLABUS CONTENTS</b>	<b>Mapped CO</b>
<b>I</b>	<b>Introduction:</b> Algorithm Analysis, Space and Time Complexity analysis, Asymptotic Notations. <b>AVL Trees</b> – Creation, Insertion, Deletion operations and Applications <b>B-Trees</b> – Creation, Insertion, Deletion operations and Applications	<b>CO1, CO3</b>
<b>II</b>	<b>Heap Trees</b> (Priority Queues) – Min and Max Heaps, Operations and Applications <b>Graphs</b> – Terminology, Representations, Basic Search and Traversals, Sets and Disjoint set Union, applications	<b>CO1, CO3</b>
<b>III</b>	<b>Divide and Conquer:</b> The General Method, Max-Min, Quick Sort, Merge Sort, Strassen's matrix multiplication <b>Greedy Method:</b> General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths	<b>CO1, CO2, CO3, CO4</b>
<b>IV</b>	<b>Dynamic Programming:</b> General Method, All pairs shortest paths, Single Source Shortest Paths– General Weights (Bellman Ford Algorithm), Optimal Binary Search Trees, 0/1 Knapsack, String Editing, Travelling Salesperson problem.	<b>CO1, CO2, CO3, CO4</b>
<b>V</b>	<b>Backtracking:</b> General Method, n-Queens Problem, Sum of Subsets problem, Graph Coloring, <b>Branch and Bound:</b> The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem.	<b>CO1, CO2, CO3, CO4</b>

	<b>Introduction to Complexity classes:</b> P and NP Problems, NP-Complete Problems.	
--	---	--

<b>Learning Resources</b>
---------------------------

<b>Text Books</b>
-------------------

- |   |
|---|
| 1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2 <sup>nd</sup> Edition Universities Press<br>2. Computer Algorithms in C++, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2 <sup>nd</sup> Edition University Press<br>3. Fundamentals of Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2 <sup>nd</sup> Edition, Universities Press, |
|---|

<b>References Text Book</b>
-----------------------------

- |  |
|--|
| 1. Data Structures and program design in C, Robert Kruse, Pearson Education Asia<br>2. An introduction to Data Structures with applications, Trembley & Sorenson, McGraw Hill<br>3. The Art of Computer Programming, Vol.1: Fundamental Algorithms, Donald E Knuth, Addison-Wesley, 1997.<br>4. Data Structures using C & C++: Langsam, Augenstein & Tanenbaum, Pearson, 1995<br>5. Fundamentals of Data Structures in C++: Horowitz Sahni & Mehta, Galgottia Pub.<br>6. Data structures in Java:, Thomas Standish, Pearson Education Asia |
|--|

<b>e-Resources and other Digital Material</b>
---

- |   |
|---|
| 1. <a href="https://www.tutorialspoint.com/advanced_data_structures/index.asp">https://www.tutorialspoint.com/advanced_data_structures/index.asp</a><br>2. <a href="http://peterindia.net/Algorithms.html">http://peterindia.net/Algorithms.html</a><br>3. Abdul Bari, <a href="#">Introduction to Algorithms (youtube.com)</a> |
|---|

**OBJECT ORIENTED PROGRAMMING THROUGH JAVA**

<b>Course Code</b>	23CS3302	<b>Year</b>	II	<b>Semester</b>	I
<b>Course Category</b>	PC	<b>Branch</b>	CSE	<b>Course Type</b>	THEORY
<b>Credits</b>	3	<b>L – T – P</b>	3-0-0	<b>Prerequisites</b>	C Programming language
<b>Continuous Evaluation:</b>	30	<b>Semester End Evaluation:</b>	70	<b>Total Marks:</b>	100

<b>Course Outcomes</b>		
Upon successful completion of the course, the student will be able to:		
<b>CO1</b>	Understand the syntax and semantics of JAVA programming language to solve a given problem.	L2
<b>CO2</b>	Apply the knowledge of Object Oriented Programming principles to develop applications.	L3
<b>CO3</b>	Apply the concepts of packages, I/O and Exception handling mechanisms to develop efficient programming.	L3
<b>CO4</b>	Analyze the concepts of Multithreading and Collection Framework to develop applications which mimic the real-world scenarios.	L4

<b>Syllabus</b>		
<b>Unit No.</b>	<b>CONTENTS</b>	<b>Mapped CO</b>
<b>I</b>	<p><b>Object Oriented Programming:</b> Basic concepts, Principles, Program Structure in JAVA: Introduction, Writing Simple JAVA Programs, Elements or Tokens in JAVA Programs, JAVA Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.</p> <p><b>Data Types, Variables, and Operators :</b>Introduction, Data Types in JAVA, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, <b>Introduction to Operators</b>, Precedence and Associativity of Operators, Assignment Operator ( = ), Basic Arithmetic Operators, Increment (++) and Decrement ( - - ) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.</p> <p><b>Control Statements :</b>Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator? :, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.</p>	<b>CO1</b>
<b>II</b>	<p><b>Classes and Objects:</b> Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another.</p> <p><b>Constructor and Methods:</b>Introduction, Defining Methods, Overloaded Methods, Constructors, Overloaded Constructor Methods , Class Objects as Parameters in Methods, Access Control, Accessing Private Members of Class, Recursive Methods, final method, Passing Arguments by Value and by Reference, Keyword this, final and static, Nested classes.</p> <p><b>String Handling in JAVA:</b> Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer.</p>	<b>CO2</b>

III	<p><b>Arrays:</b> Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors.</p> <p><b>Inheritance:</b> Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes.</p> <p><b>Interfaces:</b> Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface.</p>	CO2
IV	<p><b>Packages and JAVA Library:</b> Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in JAVA SE, java.lang Package and its Classes, Class Object, class Math, Wrapper Classes, Auto-boxing and Auto-unboxing.</p> <p><b>Exception Handling:</b> Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions, custom exceptions.</p> <p><b>Java I/O and File:</b> Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java: File, FileInputStream and FileOutputStream Classes(Text Book 2)</p>	CO3
V	<p><b>Multithreaded Programming:</b> Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Thread Priority</p> <p><b>Collections Framework :</b> Introduction,Purpose of Collection Framework Application of Collection Framework, Hierarchy of collection Interfaces / classes, Methods defined in Collection Interface, Interface Iterator, Collection classes/Interfaces –List,Queue,Set, ArrayList class, HashSet, PriorityQueue</p>	CO4

Learning Resources	
Text Books	
1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford. 2) Joy with JAVA, Fundamentals of Object Oriented Programming, DebasisSamanta, Monalisa Sarma, Cambridge, 2023.	
Reference Books	
1) The complete Reference Java, 11 <sup>th</sup> edition, Herbert Schildt, TMH 2) Introduction to Java programming, 7 <sup>th</sup> Edition, Y Daniel Liang, Pearson	
E-Resources & other digital material	
1) <a href="https://nptel.ac.in/courses/106/105/106105191/">https://nptel.ac.in/courses/106/105/106105191/</a> 2) <a href="https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview">https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview</a>	



## PYTHON PROGRAMMING

<b>Course Code</b>	<b>23SO8355</b>	<b>Year</b>	<b>II</b>	<b>Semester</b>	<b>I</b>
<b>Course Category</b>	Skill Enhancement Course	<b>Branch</b>	<b>CSE</b>	<b>Course Type</b>	<b>Practical</b>
<b>Credits</b>	2	<b>L – T – P</b>	0-1-2	<b>Prerequisites</b>	Nil
<b>Continuous Evaluation:</b>	30	<b>Semester End Evaluation:</b>	70	<b>Total Marks:</b>	100

Course Outcomes		
Upon successful completion of the course, the student will be able to:		
<b>CO1</b>	Apply Python programming constructs for solving problems.	L3
<b>CO2</b>	Conduct experiments as an individual, or team member by using Python programming.	L3
<b>CO3</b>	Develop an effective report based on various programs implemented.	L3
<b>CO4</b>	Apply technical knowledge for a given problem and express with an effective oral communication	L3
<b>CO5</b>	Analyze outputs generated through Python programming.	L4

Syllabus		
Unit No.	CONTENTS	Mapped CO
<b>I</b>	<p>UNIT-I: History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter Notebook. Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language. Control Flow Statements: if statement, if-else statement, if...elif...else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement. <b>Sample Experiments</b></p> <ol style="list-style-type: none"> <li>1. Write a Program to print the student details using Escape sequence characters.(Example:\n,\t,\").</li> <li>2. The total number of students in a class are 45 out of which 25 are boys. If 80% of the total students secured grade 'A' out of which 16 are boys, then Develop a Program to calculate the total number of girls getting grade 'A'.</li> <li>3. Develop a Program to calculate the sum of the first and the last digit of a 56743</li> <li>4. Write a program for calculating the bill amount for an item with the following scenarios <ul style="list-style-type: none"> <li>• The quantity of item sold, and price of the item must read from the user and calculate the bill</li> <li>• After that there is a 10% discount on bill amount</li> <li>• There is a tax amount of 12%</li> <li>• Find the total bill after availing the discount and applying the tax</li> </ul> </li> </ol>	<b>CO1,CO2,CO3,CO4,CO5</b>

5. Implement a program to calculate in how many days a work will be completed by three persons A, B and C together. A, B, C take x days, y days and z days respectively to do the job alone. The formula to calculate the number of days if they work together is  $xyz/(xy + yz + xz)$  days where x, y, and z are given as input to the program.
6. Implement a program to read two complex numbers and perform addition, subtraction
7. Develop a program to demonstrate evolution of following arithmetic expressions?
  - Consider b=4, c=8, d=2, e=4, f=2
  - $a=b+c/d+e*f$
  - $a=(b+c)/d+e*f$
  - $a=b+c/((d+e)*f)$
8. Write a Python program that takes two lists as input and concatenates them using the "+" operator.
9. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, and display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is  $\geq 60$  and  $< 75$ , then the grade is First Division. If aggregate is  $\geq 50$  and  $< 60$ , then the grade is Second division. If aggregate is  $\geq 40$  and  $< 50$ , then the grade is third division. Else the grade is Fail.
10. Write a program to calculate roots of a quadratic equation. The programmer has to identify whether the roots are real, equal or imaginary
11. A company decides to give bonus to all its employees on Diwali. A 5% bonus on salary is given to the male workers and 10% bonus on salary to the female workers. Write a program to enter the salary and gender of the employee. If the salary of the employee is less than Rs. 10,000 then the employee gets an extra 2% bonus on salary. Calculate the bonus that must be given to the employee and display the salary that the employee will get.
12. Demonstrate a program to print the sum of the series  $1/1^2 + 1/2^2 + 1/3^2 + \dots + 1/n^2$ , where n is taken from the user.
13. Write a program to implement the below scenarios
  - Sum of cubes of numbers from 1 to n using range ().
  - Display the numbers in descending order using range ().
  - Sum of squares of even numbers from 1 to n using range ().
  - Display all leap years from 2000 – 2200 using range ().

14. Write a program to print the below patterns:

1	1	5 4 3 2	*	
2 3	2 1	1	* *	1
4 5 6	3 2 1	4 3 2 1	* * *	1 2
7 8 9 10	4 3 2	3 2 1	* * * *	1 2 3
11 12 13 14	1	2 1	* * * *	1 2 3 4
15	5 4 3	1	*	
	2 1			

15. Create a library with functions to input the values with exception handling in Python

16. Write a Python program input and add two integers only and handle the

	exceptions.	
<b>II</b>	<p><b>UNIT-II:</b></p> <p>Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.</p> <p>Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings. Regular expression: Matching the patterns, Search and replace.</p> <p><b>Sample Experiments</b></p> <ol style="list-style-type: none"> <li>1. Write a program to find sum of all odd numbers between 1 to n using functions.</li> <li>2. Write a program to demonstrate default arguments with keyword arguments, to display name, age and salary of an employee. Where course (B. Tech) is passed as default argument.</li> <li>3. Write a program to find the sum of first 10 natural numbers using lambda or anonymous function using range () function.</li> <li>4. Demonstrate a program to convert time into minutes using functions</li> <li>5. Write a program to calculate simple interest. Suppose the customer is a senior citizen. He is being offered 12% rate of interest (ROI).For all other customers, the ROI is 10%.</li> <li>6. Python Program to check if two numbers are amicable numbers or not. Two different numbers are called amicable numbers if the sum of the proper divisors of each is equal to the other number.</li> <li>7. Demonstrate a program to sum the series <math>1/1!+4/2!+27/3+\dots n</math> using functions</li> <li>8. Write a program to generate the following pattern using default arguments. Consider four types in calling the function. <ul style="list-style-type: none"> <li>• Do not pass arguments</li> <li>• Pass only the character as argument</li> <li>• Pass character and no. of rows as argument</li> <li>• Pass character, no. of rows and columns as arguments.</li> </ul> </li> <li>9. Write a program using recursive functions: <ul style="list-style-type: none"> <li>• Counting the no. of times, a recursive function is called</li> <li>• Power of a number</li> <li>• GCD of two given numbers</li> <li>• Print the Fibonacci series</li> </ul> </li> <li>10. Write a python program without using the built in functions to find the length of the string, reverse the string.</li> <li>11. Write a python program to arrange string characters such that lowercase letters should come first.</li> <li>12. Write a program that uses regular expressions to validate dates entered by users. The program should check that the date is in a valid format, such as MM/DD/YYYY and that the month, day, and year values are within a valid range.</li> <li>13. Write a program to validate a password using regular expressions using the following rules <ul style="list-style-type: none"> <li>• At least 8 characters long</li> <li>• Contains at least one uppercase letter</li> <li>• Contains at least one lowercase letter</li> <li>• Contains at least one digit</li> </ul> </li> <li>14. Write a program to remove all non-alphanumeric characters from a given string using regular expressions.</li> </ol>	CO1,C02,C03,C04,C05

III	<p><b>UNIT-III:</b>  Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.  Dictionaries: Creating Dictionary, Accessing and Modifying key:value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement.  Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozenset.</p> <p><b>Sample Experiments</b></p> <ol style="list-style-type: none"> <li>1. Write a python program to add each element of list x with list y using nested loops.</li> <li>2. Write a python program to print index at which a particular value exists. If the value exists at multiple locations in the list, then print all the indices. Also, count the number of times that value is repeated in the list.</li> <li>3. Write a python program applying all the list methods ('append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort') on the given list.  List = [100,'a','b',102,2.3,4.5]</li> <li>4. Write a python program to add each element of x list with each element of y list. <ul style="list-style-type: none"> <li>• Using loops</li> <li>• Using list comprehension</li> </ul> </li> <li>5. Write a program using lambda and below functions to perform tasks <ul style="list-style-type: none"> <li>▪ Using filter () to filter out even numbers from a list.</li> <li>▪ Find squares of elements in a list using map ().</li> <li>▪ Product of elements of a list using reduce() function</li> </ul> </li> <li>6. Write a python program to do the below matrix operations <ul style="list-style-type: none"> <li>• Addition</li> <li>• Subtraction</li> <li>• Multiplication</li> </ul> </li> <li>7. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.</li> <li>8. Write a program to count the number of vowels in a string (No control flow allowed).</li> <li>9. Write a program to check if a given key exists in a dictionary or not.</li> <li>10. Write a program to add a new key-value pair to an existing dictionary.</li> <li>11. Write a program to sum all the items in a given dictionary.</li> <li>12. Write a program that reads string from user. Your program should create a dictionary having key as word length and value is count of words of that length. For example, if user enters 'A fat cat is on the mat'. The content of dictionary should be {1:1, 3:4, 2:2}</li> </ol>	CO1,C02,C03,C04,C05
IV	<p><b>UNIT-IV:</b>  Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.  Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.</p> <p><b>Sample Experiments</b></p>	CO1,C02,C03,C04,C05

	<ol style="list-style-type: none"> <li>1. Write a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper-case words from source must be lowered.</li> <li>2. Python program to print each line of a file in reverse order.</li> <li>3. Python program to compute the number of characters, words and lines in a file.</li> <li>4. Write a function <code>lines_count()</code> that reads lines from a text file named 'zen.txt' and displays the lines that begin with any vowel. Assume the file contains the following text and already exists on the computer's disk: Beautiful is better than ugly. Explicit is better than implicit. Simple is better than complex. Complex is better than complicated. The <code>lines_count()</code> function should display the output as: Explicit is better than implicit.</li> <li>5. Write a Python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.</li> <li>6. . Create a Parallelepiped child class inheriting from the Rectangle class and with a height attribute and another <code>Volume()</code> method to calculate the volume of the Parallelepiped.</li> <li>7. Write the complete code for BankAccount class based on the description given below: <ul style="list-style-type: none"> <li>➤ Create a Python class called BankAccount which represents a bank account, having attributes: <code>accountNumber</code> (numeric type), <code>Name</code> (name of the account owner as string type), <code>balance</code>.</li> <li>➤ Create a constructor with parameters: <code>accountNumber</code>, <code>name</code>, <code>balance</code></li> <li>➤ Create a <code>Deposit()</code> method which manages the deposit actions.</li> <li>➤ Create a <code>Withdrawal()</code> method which manages withdrawals actions.</li> <li>➤ Create a <code>bankFees()</code> method to apply the bank fees with a percentage of 5% of the balance account.</li> <li>➤ Create a <code>display()</code> method to display account details.</li> </ul> </li> </ol>	
V	<p><b>UNIT-V:</b> Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas, Matplotlib .Seaborn: Categorical Data Analysis, Regression Plots</p> <p><b>Sample Experiment</b></p> <ol style="list-style-type: none"> <li>1. Python program to check whether a JSON string contains complex object or not.</li> <li>2. Python Program to demonstrate NumPy arrays creation using <code>array ()</code> function.</li> <li>3. Python program to demonstrate use of <code>ndim</code>, <code>shape</code>, <code>size</code>, <code>dtype</code>.</li> <li>4. Python program to demonstrate basic slicing, integer and Boolean indexing.</li> <li>5. Python program to find <code>min</code>, <code>max</code>, <code>sum</code>, <code>cumulative sum</code> of array</li> <li>6. Create a dictionary with at least five keys and each key represent value as a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows: <ol style="list-style-type: none"> <li>a) Apply <code>head ()</code> function to the pandas data frame</li> <li>b) Perform various data selection operations on Data Frame</li> </ol> </li> <li>7. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attribute with scatter and plot operations in matplotlib</li> </ol>	CO1,C02,CO3,C04,C05

	<ol style="list-style-type: none"> <li>8. Create a heatmap using seaborn library showing the number of passengers over the years and months using the flights dataset.</li> <li>9. Create a simple linear regression plot using seaborn library showing the relationship between total bill and tip using the <code>tips</code> dataset.</li> </ol>	
--	---	--

Learning Resources	
Text Books	
1.	Gowrishankar S, Veena A., Introduction to Python Programming, CRC Press.
Reference Books	
1.	Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2 <sup>nd</sup> Edition, Pearson, 2024
2.	Introduction to Programming Using Python, Y. Daniel Liang, Pearson.
E-Resources & other digital material	
1.	<a href="https://www.coursera.org/learn/python-for-applied-data-science-ai">https://www.coursera.org/learn/python-for-applied-data-science-ai</a>
2.	<a href="https://www.coursera.org/learn/python?specialization=python#syllabus">https://www.coursera.org/learn/python?specialization=python#syllabus</a>

## ADVANCED DATA STRUCTURES AND ALGORITHM ANALYSIS LAB

<b>Course Code:</b>	<b>23CS3351</b>	<b>Year:</b>	<b>II</b>	<b>Semester:</b>	<b>I</b>
<b>Course Category:</b>	Professional Core Course	<b>Branch:</b>	CSE	<b>Course Type:</b>	Practical
<b>Credits:</b>	1.5	<b>L-T-P:</b>	0-0-3	<b>Prerequisites:</b>	Data Structures through C / Object Oriented Programming
<b>Continuous Internal Evaluation:</b>	30	<b>Semester End Evaluation:</b>	70	<b>Total Marks:</b>	100

## COURSE OUTCOMES

Upon successful completion of the course, Student will be able to

<b>CO1</b>	Implement programs as an individual on different IDEs/ online platforms.	<b>L3</b>
<b>CO2</b>	Apply different design techniques for solving problems.	<b>L3</b>
<b>CO3</b>	Develop an effective report based on various programs implemented.	<b>L3</b>
<b>CO4</b>	Apply technical knowledge for a given problem and express with an effective oral communication.	<b>L3</b>
<b>CO5</b>	Analyze outputs using given constraints/test cases.	<b>L4</b>

<b>Unit No.</b>	<b>SYLLABUS CONTENTS</b>	<b>Mapped CO</b>
<b>1</b>	a) Implement AVL Trees and its operations. b) Develop a solution to the given problem using AVL Trees.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>2</b>	a) Implement B- Trees and its operations. b) Develop a solution to the given problem using B- Trees.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>3</b>	a) Implement Binary Heap and its operations. b) Develop a solution to the given problem using Binary Heaps.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>4</b>	a) Implement Graph and its operations. b) Develop a solution to the given problem using Graphs.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>5</b>	Develop and implement an algorithm using Divide and Conquer strategy for a given set of problems.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>6</b>	Make use of Greedy method to implement a solution for a given problem.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>7</b>	Develop and implement an efficient solution using Dynamic Programming.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>8</b>	Use Backtracking design technique to implement a solution for a given problem.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>9</b>	Develop and implement an algorithm using Branch and Bound technique for solving a given problem.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>10</b>	<b>Case Study-1:</b> Apply the most appropriate design technique to develop and implement an efficient solution for a given problem.	<b>CO1, CO2, CO3, CO4, CO5</b>
<b>11</b>	<b>Case Study-2:</b> Develop and implement an optimal solution for a given problem by applying a suitable design technique.	<b>CO1, CO2, CO3, CO4, CO5</b>

<b>Learning Resources</b>
---------------------------

<b>Text Books</b>
-------------------

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2<sup>nd</sup> Edition Universities Press</li> <li>2. Computer Algorithms in C++, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2<sup>nd</sup> Edition University Press</li> </ol> |
|--|

<b>References Text Book</b>
-----------------------------

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Introduction to the Design &amp; Analysis of Algorithms, Anany Levitin, Third Edition, 2011, Pearson Education.</li> <li>2. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, 2002, Pearson.</li> <li>3. Algorithm Design Techniques, Narasimha Karumanchi, CareerMonk Publications, 2018.</li> </ol> |
|--|

<b>e-Resources and other Digital Material</b>
---

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. <a href="https://www.cs.usfca.edu/~galles/visualization/Algorithms.html">https://www.cs.usfca.edu/~galles/visualization/Algorithms.html</a></li> <li>2. <a href="http://littlesvr.ca/dsa-html5-animations/sorting.php">http://littlesvr.ca/dsa-html5-animations/sorting.php</a></li> <li>3. <a href="https://www.youtube.com/watch?v=AfYqN3fGapc">https://www.youtube.com/watch?v=AfYqN3fGapc</a></li> </ol> |
|--|



**OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB**

<b>Course Code</b>	23CS3352	<b>Year</b>	II	<b>Semester</b>	I
<b>Course Category</b>	PC	<b>Branch</b>	CSE	<b>Course Type</b>	PRACTICAL
<b>Credits</b>	3	<b>L – T – P</b>	0-0-3	<b>Prerequisites</b>	C Programming language
<b>Continuous Evaluation:</b>	30	<b>Semester End Evaluation:</b>	70	<b>Total Marks:</b>	100

<b>Course Outcomes</b>		
Upon successful completion of the course, the student will be able to:		
<b>CO1</b>	Implement the programs by using basics and fundamental concepts of JAVA.	L3
<b>CO2</b>	Apply the knowledge of OOP principles to develop applications.	L3
<b>CO3</b>	Analyze the Java code to write bug free programs.	L4
<b>CO4</b>	Use APIs to develop different applications in JAVA.	L3

<b>Syllabus</b>		
<b>S No.</b>	<b>CONTENTS</b>	<b>Mapped CO</b>
1	<b>Exercise – 1:</b> a) Write a JAVA program to display default value of all primitive data type of JAVA b) Write a JAVA program that display the roots of a quadratic equation $ax^2+bx=0$ . Calculate the discriminate D and basing on value of D, describe the nature of root.	CO1, CO2,CO3,CO4
2	<b>Exercise - 2</b> a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism. b) Write a JAVA program to sort for an element in a given list of elements using bubble sort c) Write a JAVA program using StringBuffer to delete, remove character.	CO1, CO2,CO3,CO4
3	<b>Exercise - 3</b> a) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method. b) Write a JAVA program implement method overloading. c) Write a JAVA program to implement constructor. d) Write a JAVA program to implement constructor overloading.	CO1, CO2,CO3,CO4
4	<b>Exercise - 4</b> a) Write a JAVA program to implement Single Inheritance b) Write a JAVA program to implement multi level Inheritance c) Write a JAVA program for abstract class to find areas of different shapes	CO1, CO2,CO3,CO4
5	<b>Exercise - 5</b> a) Write a JAVA program give example for “super” keyword. b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved? c) Write a JAVA program that implements Runtime polymorphism	CO1, CO2,CO3,CO4

6	<b>Exercise – 6</b> a) Write a JAVA program that describes exception handling mechanism b) Write a JAVA program Illustrating Multiple catch clauses c) Write a JAVA program for creation of JAVA Built-in Exceptions d) Write a JAVA program for creation of User Defined Exception	CO1, CO2,CO3,CO4
7	<b>Exercise – 7</b> a) Write a JAVA program that import and use the user defined packages. b) Write a JAVA program that import and use the user defined packages with jar file C) Write a Java Program to explore the following classes i) Formatter class ii) Random Class iii) Formatting for Date/Time in Java	CO1, CO2,CO3,CO4
8	<b>Exercise – 8</b> a) Write a JAVA program that creates threads by extending Thread class. First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds,(Repeat the same by implementing Runnable) illustrating b) Write a program <b>is Alive</b> and <b>join ()</b> c) Write a Program illustrating Daemon Threads.	CO1, CO2,CO3,CO4
9	<b>Exercise – 9</b> a) Implement the programs using ArrayList class b) Implement the programs using HashSet class c) Implement the programs using PriorityQueue class	CO1, CO2,CO3,CO4

Learning Resources	
Text Books	
1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford. 2) Joy with JAVA, Fundamentals of Object Oriented Programming, DebasisSamanta, MonalisaSarma, Cambridge, 2023.	
Reference Books	
1) The complete Reference Java, 11 <sup>th</sup> edition, Herbert Schildt, TMH 2) Introduction to Java programming, 7 <sup>th</sup> Edition, Y Daniel Liang, Pearson	
E-Resources & other digital material	
1) <a href="https://nptel.ac.in/courses/106/105/106105191/">https://nptel.ac.in/courses/106/105/106105191/</a> 2) <a href="https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview">https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview</a>	