NEC

R23 II YEAR COURSE STRUCTURE

II YEAR I SEMESTER

S.No.	Subject Code	Category	Title	L	T	P	Credits
1	R23CC2101	BS&H	Discrete Mathematics & Graph Theory	3	0	0	3
2	R23CC2102	BS&H	Universal human values – understanding harmony and Ethical human conduct	2	1	0	3
3	R23CC2106	Engineering Science	Artificial Intelligence	3	0	0	3
4	R23CC2104	Professional Core	Advanced Data Structures &Algorithms Analysis	3	0	0	3
5	R23CC2105	Professional Core	Object Oriented Programming Through Java	3	0	0	3
6	R23CC21L1	Professional Core	Advanced Data Structures and Algorithms Analysis Lab	0	0	3	1.5
7	R23CC21L2	Professional Core	Object Oriented Programming Through Java Lab	0	0	3	1.5
8	R23CC21L3	Skill Enhancement course	Python programming	0	1	2	2
		Total		14	2	8	20

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	3	0	0	30	70	100	3
SUBCODE: R23CC2101	**DISCRETE MATHEMATICS & GRAPH THEORY**						

COURSE OBJECTIVES:
- To introduce the students to the topics and techniques of discrete methods and combinatorial reasoning.
- To introduce a wide variety of applications. The algorithmic approach to the solution of problems is fundamental in discrete mathematics, and this approach reinforces the close ties between this discipline and the area of computer science.

COURSE OUTCOMES: At the end of the course students will be able to

CO1: Apply the logical statements, connectivity among the statements and different types of normal forms. [K3]

CO2: Analyze the operations, properties and functions of sets. [K4].

CO3: Solve mathematical problems with recurrence relations using different methods. [K3].

CO4: Classify the types of graphs to formulate and solve computational problems. [K4].

UNIT–I: Mathematical Logic:

Propositional Calculus: Statements and Notations, Connectives, Well Formed Formulas, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal Forms, Theory of Inference for Statement Calculus, Consistency of Premises, Indirect Method of Proof, Predicate Calculus: Predicates, Predicative Logic, Statement Functions, Variables and Quantifiers, Free and Bound Variables, Inference Theory for Predicate Calculus.

UNIT-II: Set Theory:

Sets: Operations on Sets, Principle of Inclusion-Exclusion, **Relations:** Properties, Operations, Partition and Covering, Transitive Closure, Equivalence, Compatibility and Partial Ordering, Hasse Diagrams, **Functions:** Bijective, Composition, Inverse, Permutation, and Recursive Functions, Lattice and its Properties.

UNIT-III: Combinatorics and Recurrence Relations:

Basis of Counting, Permutations, Permutations with Repetitions, Circular and Restricted Permutations, Combinations, Restricted Combinations, Binomial and Multinomial Coefficients and Theorems.

Recurrence Relations: Generating Functions, Function of Sequences, Partial Fractions, Calculating Coefficient of Generating Functions, Recurrence Relations, and Formulation as Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, Method of Characteristic Roots, Solving Inhomogeneous Recurrence Relations.

UNIT-IV: Graph Theory:
Basic Concepts, Graph Theory and its Applications, Subgraphs, Graph Representations: Adjacency and Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs.

UNIT-V: Multi Graphs:
Multigraphs, Bipartite and Planar Graphs, Euler's Theorem, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Prim's and Kruskal's Algorithms, BFS and DFS Spanning Trees.

TEXT BOOKS:
1. Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay and P. Manohar, Tata McGraw Hill.
2. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L.Liu and D. P. Mohapatra, 3rd Edition, Tata McGraw Hill.
3. Theory and Problems of Discrete Mathematics, Schaum's Outline Series, Seymour Lipschutz and Marc Lars Lipson, 3rd Edition, McGraw Hill.

REFERENCE BOOKS:
1. Discrete Mathematics for Computer Scientists and Mathematicians, J. L.Mott, A. Kandel and T. P. Baker, 2nd Edition, Prentice Hall of India publishers.
2. Discrete Mathematics, S. K. Chakraborthy and B. K. Sarkar, Oxford press, 2011.
3. Discrete Mathematics and its Applications with Combinatorics and GraphTheory, K.H. Rosen, 7th Edition, Tata McGraw Hill.

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	2	1	0	30	70	100	3
SUBCODE: R23CC2102	UNIVERSAL HUMAN VALUES UNDERSTANDING HARMONY AND ETHICAL HUMAN CONDUCT						

COURSE OBJECTIVES:

- To help the students appreciate the essential complementary between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
- To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
- To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

COURSE OUTCOMES:

At the end of the course students will be able to

CO1: Analyze various Value Education methods. [K2]

CO2: Compare and Contrast various types of Harmony in the Human Being. [K4]

CO3: Compare and Contrast various types of Harmony in the Family and Society [K4]

CO4: Compare and Contrast various types of Harmony in the Nature/Existence. [K4]

CO5: Analyze the various consequences of professional ethics. [K2]

COURSE TOPICS

- The course has 28 lectures and 14 tutorials in 5 modules. The lectures and tutorials are of 1-hour duration. Tutorial sessions are to be used to explore and practice what has been proposed during the lecture sessions.
- The Teacher's Manual provides the outline for lectures as well as practice sessions. The teacher is expected to present the issues to be discussed as propositions and encourage the students to have a dialogue.

UNIT I

Introduction to Value Education (6 lectures and 3 tutorials for practice session)

Lecture 1: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education)

Lecture 2: Understanding Value Education

Tutorial 1: Practice Session PS1 Sharing about Oneself

Lecture 3: Self-exploration as the Process for Value Education

Lecture4: Continuous Happiness and Prosperity – the Basic Human Aspirations

Tutorial 2: Practice Session PS2 Exploring Human Consciousness

Lecture 5: Happiness and Prosperity – Current Scenario

Lecture 6: Method to Fulfill the Basic Human Aspirations
Tutorial 3: Practice Session PS3 Exploring Natural Acceptance

UNIT II
Harmony in the Human Being (6 lectures and 3 tutorials for practice session)
Lecture 7: Understanding Human being as the Co-existence of the self and the body.
Lecture 8: Distinguishing between the Needs of the self and the body
Tutorial 4: Practice Session PS4 Exploring the difference of Needs of self and body.
Lecture 9: The body as an Instrument of the self
Lecture 10: Understanding Harmony in the self
Tutorial 5: Practice Session PS5 Exploring Sources of Imagination in the self
Lecture 11: Harmony of the self with the body
Lecture 12: Programme to ensure self-regulation and Health
Tutorial 6: Practice Session PS6 Exploring Harmony of self with the body

UNIT III
Harmony in the Family and Society (6 lectures and 3 tutorials for practice session)
Lecture 13: Harmony in the Family – the Basic Unit of Human Interaction
Lecture 14: 'Trust' – the Foundational Value in Relationship
Tutorial 7: Practice Session PS7 Exploring the Feeling of Trust
Lecture 15: 'Respect' – as the Right Evaluation
Tutorial 8: Practice Session PS8 Exploring the Feeling of Respect
Lecture 16: Other Feelings, Justice in Human-to-Human Relationship
Lecture 17: Understanding Harmony in the Society
Lecture 18: Vision for the Universal Human Order
Tutorial 9: Practice Session PS9 Exploring Systems to fulfil Human Goal

UNIT IV
Harmony in the Nature/Existence (4 lectures and 2 tutorials for practice session)
Lecture 19: Understanding Harmony in the Nature
Lecture 20: Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature
Tutorial 10: Practice Session PS10 Exploring the Four Orders of Nature
Lecture 21: Realizing Existence as Co-existence at All Levels
Lecture 22: The Holistic Perception of Harmony in Existence
Tutorial 11: Practice Session PS11 Exploring Co-existence in Existence.

UNIT V
Implications of the Holistic Understanding-a Look at Professional Ethics (6 lectures and 3 tutorials for practice session)
Lecture 23: Natural Acceptance of Human Values
Lecture 24: Definitiveness of (Ethical) Human Conduct
Tutorial 12: Practice Session PS12 Exploring Ethical Human Conduct
Lecture 25: A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order
Lecture 26: Competence in Professional Ethics

Tutorial 13: Practice Session PS13 Exploring Humanistic Models in Education
Lecture 27: Holistic Technologies, Production Systems and Management Models-Typical Case Studies
Lecture 28: Strategies for Transition towards Value-based Life and Profession
Tutorial 14: Practice Session PS14 Exploring Steps of Transition towards Universal Human Order

Practice Sessions for UNIT I –
Introduction to Value Education
PS1 Sharing about Oneself
PS2 Exploring Human Consciousness
PS3 Exploring Natural Acceptance

Practice Sessions for UNIT II –
Harmony in the Human Being
PS4 Exploring the difference of Needs of self and body
PS5 Exploring Sources of Imagination in the self
PS6 Exploring Harmony of self with the body

Practice Sessions for UNIT III –
Harmony in the Family and Society
PS7 Exploring the Feeling of Trust
PS8 Exploring the Feeling of Respect
PS9 Exploring Systems to fulfil Human Goal

Practice Sessions for UNIT IV –
Harmony in the Nature (Existence)
PS10 Exploring the Four Orders of Nature
PS11 Exploring Co-existence in Existence

Practice Sessions for UNIT V –
Implications of the Holistic Understanding – a Look at Professional Ethics
PS12 Exploring Ethical Human Conduct
PS13 Exploring Humanistic Models in Education
PS14 Exploring Steps of Transition towards Universal Human Order

READINGS:
Textbook and Teachers Manual
a. The Textbook
R R Gaur, R Asthana, G P Bagaria, a Foundation Course in Human Values and Professional Ethics, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1
b. The Teacher's Manual
R R Gaur, R Asthana, G P Bagaria,Teachers' Manual for A Foundation Course in Human Values and Professional Ethics, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-53-2

REFERENCE BOOKS

1. JeevanVidya: EkParichaya, A Nagaraj, JeevanVidyaPrakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj – PanditSunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)

MODE OF CONDUCT:

Lecture hours are to be used for interactive discussion, placing the proposals about the topics at hand and motivating students to reflect, explore and verify them.

Tutorial hours are to be used for practice sessions.

While analyzing and discussing the topic, the faculty mentor's role is in pointing to essential elements to help in sorting them out from the surface elements. In other words, help the students explore the important or critical elements.

In the discussions, particularly during practice sessions (tutorials), the mentor encourages the student to connect with one's own self and do self-observation, self-reflection and self-exploration.

Scenarios may be used to initiate discussion. The student is encouraged to take up "ordinary" situations rather than "extra-ordinary" situations. Such observations and their analyses are shared and discussed with other students and faculty mentor, in a group sitting. Tutorials (experiments or practical) are important for the course. The difference is that the laboratory is everyday life, and practical are how you behave and work in real life. Depending on the nature of topics, worksheets, home assignment and/or activity are included. The practice sessions (tutorials) would also provide support to a student in performing actions commensurate to his/her beliefs. It is intended that this would lead to development of commitment, namely behaving and working based on basic human values.

It is recommended that this content be placed before the student as it is, in the form of a basic foundation course, without including anything else or excluding any part of this content. Additional content may be offered in separate, higher courses. This course is to be taught by faculty from every teaching department, not exclusively by any one department.

Teacher preparation with a minimum exposure to at least one 8-day Faculty Development Program on Universal Human Values is deemed essential.

ONLINE RESOURCES:

1. https://fdp-si.aicte-india.org/UHV
   II%20Class%20Notes%20&%20Handouts/UHV%20Handout%201-
   Introduction%20to%20Value%20Education.pdf
2. https://fdp-si.aicte-india.org/UHV
   II%20Class%20Notes%20&%20Handouts/UHV%20Handout%202-
   Harmony%20in%20the%20Human%20Being.pdf
3. https://fdp-si.aicte-india.org/UHV
   II%20Class%20Notes%20&%20Handouts/UHV%20Handout%203-
   Harmony%20in%20the%20Family.pdf
4. https://fdp-si.aicte-india.org/UHV
   %201%20Teaching%20Material/D3- S2%20Respect%20July%2023.pdf
5. https://fdp-si.aicte-india.org/UHV
   II%20Class%20Notes%20&%20Handouts/UHV%20Handout%205-
   Harmony%20in%20the%20Nature%20and%20Existence.pdf
6. https://fdp-si.aicte-india.org/download/FDPTeachingMaterial/3days%20FDP-
   SI%20UHV%20Teaching%20Material/Day%203%20Handouts/UHV%203D%20D3-
   S2A%20Und%20Nature-Existence.pdf
7. https://fdp-si.aicte-
   india.org/UHV%20II%20Teaching%20Material/UHV%20II%20Lecture%2023-
   25%20Ethics%20v1.pdf
8. https://www.studocu.com/in/document/kiet-group-of-institutions/universal-human
   values/chapter-5- holistic-understanding-of-harmony-on-professional-ethics/62490385
9. https://onlinecourses.swayam2.ac.in/aic22_ge23/preview

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

NEC

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	3	0	0	30	70	100	3
SUBCODE: R23CC2106	ARTIFICIAL INTELLIGENCE						

SUBCODE: R23CC2106	ARTIFICIAL INTELLIGENCE

PRE-REQUISITE:
- Knowledge in Computer Programming.
- A course on "Mathematical Foundations of Computer Science".
- Background in linear algebra, data structures and algorithms, and probability.

COURSE OBJECTIVES:
- The student should be made to study the concepts of Artificial Intelligence.
- The student should be made to learn the methods of solving problems using Artificial Intelligence
- The student should be made to learn the methods of solving problems using Artificial Intelligence
- The student should be made to introduce the concepts of Expert Systems.
- To understand the applications of AI, namely game playing, theorem proving, and machine learning.
- To learn different knowledge representation techniques

COURSE OUTCOMES:
At the end of the course students will be able to
- **CO1:** Analyze and Design Intelligent Agents **[K4]**
- **CO2:** Apply Search Algorithms to Problem Solving **[K3]**
- **CO3:** Apply techniques for constraint propagation and reasoning under uncertainty **[K3]**
- **CO4:** Utilize inductive learning, decision trees, and explanation-based learning for learning from observation **[K3]**
- **CO5:** Analyze and compare typical expert systems such as MYCIN, DART, and XCON **[K4]**

UNIT - I
Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT - II
Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT - III

Representation of Knowledge: Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes' probabilistic interferences and dempstershafer theory.

UNIT - IV

Logic concepts: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT - V

Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON:
Expert systems shells.

TEXT BOOKS:

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", SecondEdition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", Mc Graw Hill

REFERENCE BOOKS:

1. David Poole, Alan Mackworth, Randy Goebel,"Computational Intelligence: a logical approach", Oxford University Press.
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problemsolving", Fourth Edition, Pearson Education.
3. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

ONLINE LEARNING RESOURCES:

1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	3	0	0	30	70	100	3
SUBCODE: R23CC2104	**ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS**						

COURSE OBJECTIVES:

The main objectives of the course is to
- provide knowledge on advance data structures frequently used in Computer Science domain
- Develop skills in algorithm design techniques popularly used
- Understand the use of various data structures in the algorithm design

COURSE OUTCOMES:

At the end of the course students will be able to

CO1: Apply asymptotic notations to measure the performance of algorithms [K3]

CO2: Appling divide and conquer parading when an algorithmic design situation calls for it. [K3]

CO3: Construct greedy algorithms and dynamic programming techniques to solve problems [K3]

CO4: Utilize backtracking and branch and bound algorithms to solve combinatorial problems construct [K3]

CO5: Classifying computational problems into NP, NP-Hard, and NP-Complete. [K4]

UNIT – I:

Introduction to Algorithm Analysis, Space and Time Complexity analysis, Asymptotic Notations. AVL Trees – Creation, Insertion, Deletion operations and Applications
Heap Trees (Priority Queues) – Min and Max Heaps, Operations and Applications

UNIT – II:

Graphs – Terminology, Representations, Basic Search and Traversals, Connected Components and Biconnected Components, applications Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication.

UNIT – III:

Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths
Dynamic Programming: General Method, All pairs shortest paths, Optimal Binary Search Trees, 0/1 Knapsack, String Editing, Travelling Salesperson problem

UNIT – IV:

Backtracking: General Method, 8-Queens Problem, Sum of Subsets problem, Graph Coloring, 0/1 Knapsack Problem

Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem

UNIT – V:

NP Hard and NP Complete Problems: Basic Concepts

NP Hard Graph Problems: Clique Decision Problem (CDP), Traveling Salesperson Decision Problem (TSP)

NP Hard Scheduling Problems: Job Shop Scheduling

TEXTBOOKS:

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

REFERENCE BOOKS:

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

ONLINE LEARNING RESOURCES:

1. https://www.tutorialspoint.com/advanced_data_structures/index.asp
2. http://peterindia.net/Algorithms.html
3. Abdul Bari, 1. Introduction to Algorithms (youtube.com)

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	3	0	0	30	70	100	3
SUBCODE: R23CC2105	OBJECT-ORIENTED PROGRAMMING THROUGH JAVA						

COURSE OBJECTIVES:

The learning objectives of this course are to:
- Identify Java language components and how they work together in applications
- Learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- Learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
- Understand how to design applications with threads in Java
- Understand how to use Java APIs for program development

COURSE OUTCOMES:

After completion of this course, the students would be able to:

CO1: Interpret the syntax and semantics of java programs language and OOPs concepts. [K2]

CO2: Make use of different predefined classes and packages and interfaces to develop programs using OOPs concepts. [K3]

CO3: Apply exception handling and FILE I/O operations on java programs. [K3]

CO4: Make use of Multithreading and String handling Functions on java. [K3]

CO5: Make use of Java FX and Event-Handling to in the design of GUI Applications. [K3]

UNIT I:

Object Oriented Programming: 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.

Data Types, Variables, and Operators :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, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (- -) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: 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.

UNIT II:

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing

Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.
Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

UNIT III:
Arrays: 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.
Inheritance: 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, Interfaces and Inheritance.
Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT IV:
Packages and Java Library: 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, Enumeration, class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.
Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.
Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java (Text Book 2)

UNIT V:
String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer.
Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Thread Priority-Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.
Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events (Text Book 3)

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.
3. JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

REFERENCES BOOKS:
1. The complete Reference Java, 11thedition, Herbert Schildt,TMH
2. Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson

ONLINE RESOURCES:
1. https://nptel.ac.in/courses/106/105/106105191/
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618 816347_shared/overview

NEC

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	0	0	3	30	70	100	1.5
SUBCODE: R23CC21L1	**ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS LAB**						

COURSE OBJECTIVES:
The objectives of the course is to
- Acquire practical skills in constructing and managing Data structures
- Apply the popular algorithm design methods in problem-solving scenarios

EXPERIMENTS COVERING THE TOPICS:
- Operations on AVL trees, B-Trees, Heap Trees
- Graph Traversals
- Sorting techniques
- Minimum cost spanning trees
- Shortest path algorithms
- 0/1 Knapsack Problem
- Travelling Salesperson problem
- Optimal Binary Search Trees
- N-Queens Problem
- Job Sequencing

COURSE OUTCOMES:
After the completion of this course the student should be able to
CO 1: Analyze different operations of tree traversal techniques. [K4]
CO 2: Analyze time complexity of algorithms to solve problems on graph [K4]
CO 3: Apply divide and conquer approaches for sorting the given elements [K3]
CO 4: Analyze the complexity of algorithms to evaluate the efficiency and effectiveness of greedy, dynamic programming, backtracking branch and bound techniques. [K4]

SAMPLE PROGRAMS:
1. Construct an AVL tree for a given set of elements which are stored in a file. And implement insert and delete operation on the constructed tree. Write contents of tree into a new file using in-order.
2. Construct B-Tree an order of 5 with a set of 100 random elements stored in array. Implement searching, insertion and deletion operations.
3. Construct Min and Max Heap using arrays, delete any element and display the content of the Heap.
4. Implement BFT and DFT for given graph, when graph is represented by
   a) Adjacency Matrix   b) Adjacency Lists
5. Write a program for finding the bi-connected components in a given graph.
6. Implement Quick sort and Merge sort and observe the execution time for various input sizes (Average, Worst and Best cases).

7. Compare the performance of Single Source Shortest Paths using Greedy method when the graph is represented by adjacency matrix and adjacency lists.
8. Implement Job sequencing with deadlines using Greedy strategy.
9. Write a program to solve 0/1 Knapsack problem Using Dynamic Programming.
10. Implement N-Queens Problem Using Backtracking.
11. Use Backtracking strategy to solve 0/1 Knapsack problem.
12. Implement Travelling Sales Person problem using Branch and Bound approach.

REFERENCE BOOKS:

1. Fundamentals of Data Structures in C++, Horowitz Ellis, SahniSartaj, Mehta, Dinesh, 2^{nd}Edition, Universities Press
2. Computer Algorithms/C++ Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, 2^{nd}Edition, University Press
3. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
4. An introduction to Data Structures with applications, Trembley& Sorenson, McGraw Hill

ONLINE LEARNING RESOURCES:

1. http://cse01-iiith.vlabs.ac.in/
2. http://peterindia.net/Algorithms.html

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	0	0	3	30	70	100	1.5
SUBCODE: R23CC21L2	**OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB**						

COURSE OBJECTIVES:

The aim of this course is to

- Practice object oriented programming in the Java programming language
- Implement Classes, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrate inheritance, Exception handling mechanism, JDBC connectivity
- Construct Threads, Event Handling, implement packages, Java FX GUI

EXPERIMENTS COVERING THE TOPICS:

- Object Oriented Programming fundamentals- data types, control structures
- Classes, methods, objects, Inheritance, polymorphism,
- Exception handling, Threads, Packages, Interfaces
- Files, I/O streams, JavaFX GUI

COURSE OUTCOMES:

At the end of the course students will be able to

CO1: Develop Java program, by using OOP concepts. [K3]
CO2: Make use of inheritance and interface concepts in Java programs. [K3]
CO3: Make use of Exception handling and Multithreading concepts in Java Programs. [K3]
CO4: Develop GUIs with JavaFX and JDBC programs. [K3]

SAMPLE EXPERIMENTS:

Exercise – 1:

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.

Exercise - 2

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.

Exercise - 3

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.

Exercise - 4

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

Exercise - 5

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

Exercise - 6

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

Exercise - 7

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)
b) Write a program illustrating is Alive and join ()
c) Write a Program illustrating Daemon Threads.
d) Write a JAVA program Producer Consumer Problem

Exercise – 8

a) Write a JAVA program that import and use the user defined packages
b) Without writing any code, build a GUI that display text in label and image in an ImageView (use JavaFX)
c) Build a Tip Calculator app using several JavaFX components and learn how to respond to user interactions with the GUI

Exercise – 9

a) Write a java program that connects to a database using JDBC
b) Write a java program to connect to a database using JDBC and insert values into it.
c) Write a java program to connect to a database using JDBC and delete values from it

Virtual Lab: http://ps-iiith.vlabs.ac.in/, www.w3schools.com

Any three programs must be submitted with result from the above link.

II B.TECH I SEMESTER	L	T	P	INTERNAL MARKS	EXTERNAL MARKS	TOTAL MARKS	CREDITS
	0	1	2	30	70	100	2
SUBCODE: R23CC21L3	**PYTHON PROGRAMMING (SKILL ENHANCEMENT COURSE)**						

COURSE OBJECTIVES:

The main objectives of the course are to

- Introduce core programming concepts of Python programming language.
- Demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries
- Implement Functions, Modules and Regular Expressions in Python Programming and to create practical and contemporary applications using these

COURSE OUTCOMES:

At the end of the course students will be able to

CO1: Make use of control flow statements and functions to develop python programs.[K3].

CO2: Develop Python programs using strings, Lists, dictionaries, tuples and sets. [K3].

CO3: Develop Python programs on object oriented programming and regular expressions. [K3].

CO4: Develop Python programs using Nuimpy and Pandas. [K3].

UNTI-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.

Sample Experiments:

1. Write a program to find the largest element among three Numbers.
2. Write a Program to display all prime numbers within an interval
3. Write a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.
   i) Arithmetic Operators ii) Relational Operators iii) Assignment Operators iv) Logical Operators v) Bit wise Operators vi) Ternary Operator vii) Membership Operators viii) Identity Operators
5. Write a program to add and multiply complex numbers
6. Write a program to print multiplication table of a given number.

UNIT-II:

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.

Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.

Sample Experiments:

1. Write a program to define a function with multiple return values.
2. Write a program to define a function using default arguments.
3. Write a program to find the length of the string without using any library functions.
4. Write a program to check if the substring is present in a given string or not.
5. Write a program to perform the given operations on a list:
    i. Addition   **ii.** Insertion   **iii.** Slicing
6. Write a program to perform any 5 built-in functions by taking any list.

UNIT-III:

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.

Sample Experiments:

1. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
2. Write a program to count the number of vowels in a string (No control flow allowed).
3. Write a program to check if a given key exists in a dictionary or not.
4. Write a program to add a new key-value pair to an existing dictionary.
5. Write a program to sum all the items in a given dictionary.

UNIT-IV:

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.

Sample Experiments:
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.
2. Python program to print each line of a file in reverse order.
3. Python program to compute the number of characters, words and lines in a file.
4. Write a program to create, display, append, insert and reverse the order of the items in the array.
5. Write a program to add, transpose and multiply two matrices.
6. 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.

UNIT-V:
Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas.

Sample Experiments:
1. Python program to check whether a JSON string contains complex object or not.
2. Python Program to demonstrate NumPy arrays creation using array () function.
3. Python program to demonstrate use of ndim, shape, size, dtype.
4. Python program to demonstrate basic slicing, integer and Boolean indexing.
5. Python program to find min, max, sum, cumulative sum of array
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:
   1. Apply head () function to the pandas data frame
   2. Perform various data selection operations on Data Frame
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

REFERENCE BOOKS:
1. Gowrishankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2ndEdition, Pearson, 2024
3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

ONLINE LEARNING RESOURCES / VIRTUAL LABS:
1. https://www.coursera.org/learn/python-for-applied-data-science-ai
2. https://www.coursera.org/learn/python?specialization=python#syllabus