KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology
COURSE STRUCTURE & SYLLABUS – II B. TECH (IT) – R23 Regulation
(Applicable from the academic year: 2023-24 onwards)

B.Tech.– II Year I Semester (SEMESTER-III)

S. No	Course Code	Course Title	L	T	P	C	IM	EM	TM
1	BS&H	Discrete Mathematics & Graph Theory	3	0	0	3	30	70	100
2	PC	Object Oriented Programming Through Java	3	0	0	3	30	70	100
3	ES	Digital Logic & Computer Organization	3	0	0	3	30	70	100
4	PC	Advanced Data Structures & Algorithm Analysis	3	0	0	3	30	70	100
5	BS&H	Universal Human Values– Understanding Harmony	2	1	0	3	30	70	100
6	PC	Advanced Data Structures & Algorithm Analysis Lab	0	0	3	1.5	30	70	100
7	PC	Object Oriented Programming Through Java Lab	0	0	3	1.5	30	70	100
8	Skill EC	Basic Web Development	0	1	2	2		50	50
9	AC	Environmental Science	2	0	0	-	-	-	-
Total Credits						**20**	**210**	**540**	**750**

Theory: BS&H-2, ES-1, PC-2, AC-1 **Practical:** PC-2, SEC-1

B.Tech.– II Year II Semester (SEMESTER-IV)

S. No	Course Code	Course Title	L	T	P	C	IM	EM	TM
1	Management Course	Optimization Techniques	2	0	0	2	30	70	100
2	ES / BS	Probability & Statistics	3	0	0	3	30	70	100
3	PC	Operating Systems	3	0	0	3	30	70	100
4	PC	Database Management Systems	3	0	0	3	30	750	100
5	PC	Python Programming	3	0	0	3	30	70	100
6	PC	Database Management Systems Lab	0	0	3	1.5	30	70	100
7	PC	Python Programming Lab	0	0	3	1.5	30	70	100
8	Skill EC	Full Stack Development	0	1	2	2	-	50	50
9	BS & H	Design Thinking & Innovation	0	0	2	1	-	50	50
Total Credits						**20**	**210**	**590**	**800**
Mandatory Community Service Project Internship of 08 weeks duration during summer Vacation									

Theory: ES / BS-1, PC-3, BS & H-1,MgC-1 **Practical:** PC-2, SEC-1

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year I Semester

L	T	P	C
3	0	0	3

DISCRETE MATHEMATICS & GRAPH THEORY

Course Objectives:

The learning objectives of this course are to:

- Introduce the students to the topics and techniques of discrete methods and combinatorial reasoning.
- 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

- **CO 1:** build skills in solving mathematical problems
- **CO 2:** comprehend mathematical principles and logic
- **CO 3:** demonstrate knowledge of mathematical modeling and proficiency in using mathematical software
- **CO 4:** manipulate and analyze data numerically and/or graphically using appropriate Software
- **CO 5:** how to communicate effectively mathematical ideas/results verbally or in writing

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, Formulation as Recurrence Relations, Solving

Recurrence Relations by Substitution and Generating Functions, Method of Characteristic

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.
2. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby and Sharon Cutler Ross, PHI.
3. Discrete Mathematics, S. K. Chakraborthy and B.K. Sarkar, Oxford, 2011.
4. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, K. H. Rosen, 7th Edition, Tata McGraw Hill.

Online Resources:

1. https://mu.ac.in/wp-content/uploads/2021/05/Decision-Making-and-Mathematical-Modeling-Final-1-converted.pdf
2. https://byjus.com/maths/basics-set-theory/
3. https://www.tutorialspoint.com/discrete_mathematics/discrete_mathematics_recurrence_relation.htm

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year I Semester

L	T	P	C
3	0	0	3

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:

At the end of the course, the student will be able to

- **CO 1:** realize the concept of Object-Oriented Programming & Java Programming Constructs
- **CO 2:** describe the basic concepts of Java such as operators, classes, objects, inheritance, packages, Enumeration and various keywords
- **CO 3**: apply the concept of interfaces and inheritance
- **CO 4:** design the applications of packages exception handling and Input/ Output operations and multi-threading
- **CO 5:** analyze & design the concept of Event Handling and Abstract Window Toolkit

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 print() 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

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)

Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads,

UNIT V

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer.

Event Handling: Two Event Handling Mechanisms, The Delegation Event Model, Event Classes, The Key Event Class, Sources of Events, Event Listener Interfaces, Using the Delegation Event Model.

Introducing the AWT: AWT Classes, Window Fundamentals, Working with Frame Windows, Introducing Graphics**,**

Using AWT Controls, Layout Managers, and Menus: AWT Control Fundamentals, Labels, Using Buttons, Applying Check Boxes, Checkbox Group, Choice Controls, Using Lists, Managing Scroll Bars, using a TextField, using a TextArea, Understanding Layout Managers, Menu Bars and Menus.

Text Books:

1) The complete Reference Java, 11th edition, Herbert Schildt, TMH
2) Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis

Samanta, Monalisa Sarma, Cambridge, 2023.
3) JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

References Books:

1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
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_012880464547618816347_shared/overview

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year I Semester

L	T	P	C
3	0	0	3

DIGITAL LOGIC & COMPUTER ORGANIZATION

Course Objectives:

The main objectives of the course are to

- Provide students with a comprehensive understanding of digital logic design principles and computer organization fundamentals
- Describe memory hierarchy concepts
- Explain input/output (I/O) systems and their interaction with the CPU, memory, and peripheral devices.

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** represent numbers, base conversions and simplify the complex logic functions using postulates, theorems and k-maps
- **CO 2:** design combinational circuits for various digital applications
- **CO 3:** understand the basic components and the design of CPU, instruction set, instruction formats and addressing modes
- **CO 4:** analyze and understand the memory organization
- **CO 5:** analyze and understand the input-output organization

UNIT – I:

Data Representation: Binary Numbers, Fixed Point Representation. Floating Point Representation. Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes

Digital Logic Circuits-I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions. K-Map Simplification, Combinational Circuits, Decoders, Multiplexers

UNIT – II:

Digital Logic Circuits-II: Sequential Circuits, Flip-Flops, Binary counters, Registers, Shift Registers, Ripple counters

Basic Structure of Computers: Computer Types, Von- Neumann Architecture, Basic operational concepts, Bus structures, Computer Generations, register transfer language, Bus & Memory transfer, Arithmetic, logical and shift micro-operations.

UNIT – III:

Computer Arithmetic: Addition and Subtraction of Signed Numbers, unsigned numbers adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Booths, Multiplication, Integer Division, Floating-Point Numbers and Operations

Processor Organization: Fundamental Concepts, Execution of a Complete Instruction, Addressing modes.

UNIT – IV:

The Memory Organization: Basic Concepts, Semiconductor RAM Memories, Read-Only

Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Secondary Storage

UNIT – V:

Input/Output Organization: Accessing I/O Devices, Interrupts, Memory mapped I/O, I/O mapped I/O, Asynchronous data transfer, Modes of Transfer, Direct Memory Access.

Textbooks:

1. Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, 6th edition, McGraw Hill
2. Digital Design, 6th Edition, M. Morris Mano, Pearson Education.
3. Computer Organization and Architecture, William Stallings, 11thEdition, Pearson.

Reference Books:

1. Computer Systems Architecture, M.Moris Mano, 3rdEdition, Pearson
2. Computer Organization and Design, David A. Paterson, John L.Hennessy, Elsevier
3. Fundamentals of Logic Design, Roth, 5thEdition, Thomson

Online Resources:

1. https://nptel.ac.in/courses/106/103/106103068/
2. https://nptel.ac.in/courses/106108099

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year I Semester

L	T	P	C
3	0	0	3

ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS

Course Objectives:
The main objectives of the course are to:

- Analyze performance of algorithms.
- Understand and choose the appropriate algorithm design technique for a specified application.
- Solve problems using algorithm design techniques such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.
- Analyze the impact of algorithm design techniques on each application solved.
- Introduce and understand P and NP classes.

Course Outcomes:
At the end of the course, the student will be able to

- **CO 1:** Understand the basics of algorithm analysis and analyze the performance of operations on AVL trees and B-trees.
- **CO 2:** Apply advanced trees and graphs for solving problems effectively.
- **CO 3:** Understand the use of algorithm design strategies: divide and conquer, greedy methods.
- **CO 4:** Apply dynamic programming approach to solve suitable problems.
- **CO 5:** Use state space search approaches for solving relevant problems and understand the limitations of algorithms.

UNIT – I:
Introduction to Algorithm Analysis: Space and Time Complexity analysis, Asymptotic Notations.
AVL Trees – Creation, Insertion, Deletion operations and applications
B-Trees – Creation, Insertion, Deletion operations and Applications, B+ Trees.

UNIT – II:
Heap Trees (Priority Queues) – Min and Max Heaps, Operations and Applications
Graphs – Terminology, Representations, Basic Search and Traversals, Connected Components and Biconnected Components, applications.

UNIT – III:
Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication
Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths.

UNIT – IV:
Dynamic Programming: General Method, all pairs shortest paths, Single Source Shortest

Paths– General Weights (Bellman Ford Algorithm), Optimal Binary Search Trees, 0/1 Knapsack, Travelling Salesperson problem.

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

UNIT – V:

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

NP Hard and NP Complete Problems: Basic Concepts

NP Hard Graph Problems: Clique Decision Problem (CDP), Chromatic Number Decision Problem (CNDP)

Textbooks:

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

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

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year I Semester

L	T	P	C
2	1	0	3

UNIVERSAL HUMAN VALUES – UNDERSTANDING HARMONY AND ETHICAL HUMAN CONDUCT

Course Objectives:

The main objectives of the course are:

- 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 behaviors and mutually enriching interaction with Nature.

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** define the terms like Natural Acceptance, Happiness and Prosperity
- **CO 2:** identify one's self, and one's surroundings (family, society nature)
- **CO 3:** apply what they have learnt to their own self in different day-to-day settings in real life
- **CO 4:** relate human values with human relationship and human society.
- **CO 5:** justify the need for universal human values and harmonious existence

UNIT I

Introduction to Value Education: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education), Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity, Method to Fulfil the Basic Human Aspirations

UNIT II

Harmony in the Human Being: Understanding Human being as the Co-existence of the self and the body, distinguishing between the Needs of the self and the body, Harmony of the self with the body, Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail, Programs to ensure Sanyam and Health.

UNIT III

Harmony in the Family and Society: Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, Understanding the meaning of Respect, Difference between respect and differentiation, the other salient values in relationship, Understanding Harmony in the Society.

UNIT- IV

Harmony in the Nature/Existence: Understanding Harmony in the Nature, Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence.

UNIT V

Implications of the Holistic Understanding – a Look at Professional Ethics: Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics, Production Systems and Management Models.

READINGS:

Textbook and Teachers Manual

a. The Textbook

1. 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

2. 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. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, 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)

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year I Semester

L	T	P	C
0	0	3	1.5

ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS LAB

Course Objectives:

The objectives of the course are to

- acquire practical skills in constructing and managing Data structures
- apply the popular algorithm design methods in problem-solving scenarios

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** understand and apply data structure operations.
- **CO 2:** understand and apply non-linear data structure operations.
- **CO 3:** apply Greedy, divide and conquer algorithms.
- **CO 4:** develop dynamic programming algorithms for various real-time applications.
- **CO 5:** illustrate and apply backtracking algorithms, further able to understand non-deterministic algorithms.

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

Experiments:

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 biconnected 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, 2nd Edition, Universities Press
2. Computer Algorithms/C++ Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, 2nd 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 Resources:

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

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

L	T	P	C
0	0	3	1.5

II Year I Semester

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

Course Objectives:

The objectives of this course are 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

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** write the programs for solving real world problems using Java OOP principles
- **CO 2:** write programs using Exceptional Handling approach
- **CO 3:** write multithreaded applications
- **CO 4:** build application using Java Collection Framework
- **CO 5:** develop java application using packages

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, Java FX GUI

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 String Buffer 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 implements method overloading.

c) Write a JAVA program to implement constructor.

d) Write a JAVA program to implement constructor overloading.

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

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) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Exercise – 9

Write a Java Program That works as a simple calculator using Grid layout to arrange buttons for the digits and +, -, * % operations. Add a text filed to print the result.

Text Books:

2) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
3) Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
4) JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

References Books:

3) The complete Reference Java, 11thedition, Herbert Schildt, TMH
4) Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson

Online Resources:

1. https://www.javatpoint.com/java-program-to-solve-quadratic-equation
2. https://www.programiz.com/java-programming/examples/quadratic-roots-equation

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information TechnologyII

Year I Semester

L	T	P	C
0	1	2	2

Basic Web Development (SKILL ENHANCEMENT COURSE-1)

Course Objectives:

The main objectives of the course are to

- Make use of HTML elements and their attributes for designing static web pages
- Build a web page by applying appropriate CSS styles to HTML elements
- Experiment with JavaScript to develop dynamic web pages and validate forms

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** understand the basics of full stack web development by Develop responsive web pages using HTML and CSS.
- **CO 2:** implement client-side scripting using JavaScript and Build interactive web pages.
- **CO 3:** deploy web applications on GitHub
- **CO 4:** understand the best practices for building secure web applications
- **CO 5:** develop real-world web applications using various technologies learned in the course

Experiments covering the Topics:

- Lists, Links and Images
- HTML Tables, Forms and Frames
- HTML 5 and Cascading Style Sheets, Types of CSS
- Selector forms
- CSS with Color, Background, Font, Text and CSS Box Model
- Applying JavaScript - internal and external, I/O, Type Conversion
- JavaScript Conditional Statements and Loops, Pre-defined and User-defined Objects

Sample Experiments:

1. Lists, Links and Images

a. Write a HTML program, to explain the working of lists.
Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.

b. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target Attributes.

c. Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.

d. Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique

2. HTML Tables, Forms and Frames

a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>,

<td> and attributes: border, rowspan, colspan)

a. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).
b. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select>&<option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).
c. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame → image, second frame → paragraph, third frame → hyperlink. And also make sure of using "no frame" attribute such that frames to be fixed).

3. HTML 5 and Cascading Style Sheets, Types of CSS

a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, , tags.
b. Write a HTML program, to embed audio and video into HTML web page.
c. Write a program to apply different types (or levels of styles or style specification formats)
   - inline, internal, external styles to HTML elements. (identify selector, property and value).

4. Selector forms

a. Write a program to apply different types of selector forms
   i. Simple selector (element, id, class, group, universal)
   ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
   iii. Pseudo-class selector
   iv. Pseudo-element selector
   v. Attribute selector

5. CSS with Color, Background, Font, Text and CSS Box Model

a. Write a program to demonstrate the various ways you can reference a color in CSS.
b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.
c. Write a program using the following terms related to CSS font and text:
   i. font-size ii. font-weight iii. font-style
   iv. text-decorationv. text-transformation vi. text-alignment
d. Write a program, to explain the importance of CSS Box model using
   i. Content ii. Border iii. Margin iv. padding

6. Applying JavaScript - internal and external, I/O, Type Conversion

a. Write a program to embed internal and external JavaScript in a web page.
b. Write a program to explain the different ways for displaying output.
c. Write a program to explain the different ways for taking input.

7. Java Script Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.

b. Write a program using window object properties and methods.
c. Write a program using array object properties and methods.
d. Write a program using math object properties and methods.
e. Write a program using string object properties and methods.
f. Write a program using regex object properties and methods.
g. Write a program using date object properties and methods.
h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

8. Java Script Conditional Statements and Loops

a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words "LARGER NUMBER" in an information message dialog. If the numbers are equal, output HTML text as "EQUAL NUMBERS".
b. Write a program to display week days using switch case.
c. Write a program to print 1 to 10 numbers using for, while and do-while loops.
d. Write aprogram to print data in object using for-in, for-each and for-of loops
e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]
f. Write a program to display the denomination of the amount deposited in the bank in terms of 100's, 50's, 20's, 10's, 5's, 2's & 1's. (Eg: If deposited amount is Rs.163, the output should be 1-100's, 1-50's, 1- 10's, 1-2's & 1-1's)

Text Books:

1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.Web Programming with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, 2019 (Chapters 1-11).
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.

Online Resources:

1. https://www.w3schools.com/html
2. https://www.w3schools.com/css
3. https://www.w3schools.com/js/
4. https://www.w3schools.com/nodejs

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCE
(AUTONOMOUS)

B. Tech in Information Technology

II Year I Semester

L	T	P	C
2	0	0	0

ENVIRONMENTAL SCIENCE

Course Objectives:

- To make the students to get awareness on environment
- To understand the importance of protecting natural resources, ecosystems for future generations and pollution causes due to the day-to-day activities of human life
- To save earth from the inventions by the engineers

.

Course Outcomes :

CO 1: Understand multi disciplinary nature of environmental studies and various renewable and non-renewable resources.

CO 2 Understand flow and bio-geo- chemical cycles and ecological pyramids

CO 3: Understand various causes of pollution and solid waste management and related preventive Measures

CO 4 : Understand the rainwater harvesting, watershed management, ozone layer depletion and waste land reclamation

CO 5: Ilustrate the causes of population explosion, value education and welfare programmes

UNIT – I

Multidisciplinary Nature of Environmental Studies: – Definition, Scope and Importance – Need for Public Awareness.

Natural Resources : Renewable and non-renewable resources – Natural resources and associated problems – Forest resources – Use and over – exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems – Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies – Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies. – Energy resources:

UNIT – II

Ecosystems: Concept of an ecosystem. – Structure and function of an ecosystem – Producers, consumers and decomposers – Energy flow in the ecosystem – Ecological succession – Food chains, food webs and ecological pyramids – Introduction, types, characteristic features, structure and function of the following ecosystem:

a. Forest ecosystem.
b. Grassland ecosystem
c. Desert ecosystem
d. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Biodiversity and Its Conservation : Introduction and Definition: genetic, species and ecosystem diversity – Bio-geographical classification of India – Value of biodiversity: consumptive use, Productive use, social, ethical, aesthetic and option values – Biodiversity at global, National and local levels – India as a mega-diversity nation – Hot-sports of biodiversity – Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts – Endangered and endemic species of India – Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

UNIT – III

Environmental Pollution: Definition, Cause, effects and control measures of:

a. Air Pollution.
b. Water pollution
c. Soil pollution
d. Marine pollution
e. Noise pollution
f. Thermal pollution
g. Nuclear hazards

Solid Waste Management: Causes, effects and control measures of urban and industrial wastes – Role of an individual in prevention of pollution – Pollution case studies – Disaster management: floods, earthquake, cyclone and landslides.

UNIT – IV

Social Issues and the Environment: From Unsustainable to Sustainable development – Urban problems related to energy – Water conservation, rain water harvesting, watershed management – Resettlement and rehabilitation of people; its problems and concerns. Case studies – Environmental ethics: Issues and possible solutions – Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case Studies – Wasteland reclamation. – Consumerism and waste products. – Environment Protection Act. – Air (Prevention and Control of Pollution) Act. – Water (Prevention and control of Pollution) Act – Wildlife Protection Act – Forest Conservation Act – Issues involved in enforcement of environmental legislation – Public awareness.

UNIT – V

Human Population And The Environment: Population growth, variation among nations. Population explosion – Family Welfare Programmes. – Environment and human health – Human Rights – Value Education – HIV/AIDS – Women and Child Welfare – Role of information Technology in Environment and human health – Case studies.

Field Work: Visit to a local area to document environmental assets River/forest

grassland/hill/mountain – Visit to a local polluted site-Urban/Rural/Industrial/Agricultural Study of common plants, insects, and birds – river, hill slopes, etc.

Textbooks:

1. Erach Bharucha,Text book of Environmental Studies for Undergraduate Courses,Universities Press (India) Private Limited, 2019.
2. Palaniswamy, Environmental Studies, 2/e, Pearson education, 2014.
3. S.Azeem Unnisa, Environmental Studies, Academic Publishing Company, 2021.
4. K.Raghavan Nambiar, "Text book of Environmental Studies for Undergraduate Courses as per UGC model syllabus", SciTech Publications (India), Pvt. Ltd, 2010.

Reference Books:

1. Deeksha Dave and E.Sai Baba Reddy, Textbook of Environmental Science, 2/e, Cengage Publications, 2012.
2. M.Anji Reddy, "Textbook of Environmental Sciences and Technology", BS Publication, 2014.
3. J.P. Sharma, Comprehensive Environmental studies, Laxmi publications, 2006.
4. J. Glynn Henry and Gary W. Heinke, Environmental Sciences and Engineering, Prentice Hall of India Private limited, 1988.
5. G.R. Chatwal, A Text Book of Environmental Studies, Himalaya Publishing House, 2018.
6. Gilbert M. Masters and Wendell P. Ela, Introduction to Environmental Engineering and Science, 1/e, Prentice Hall of India Private limited, 1991.

Online Learning Resources:

- https://onlinecourses.nptel.ac.in/noc23_hs155/preview
- https://www.edx.org/learn/environmental-science/rice-university-ap-r- environmental-science-part-3-pollution-and- resources?index=product&objectID=course-3a6da9f2-d84c-4773-8388-1b2f8f6a75f2&webview=false&campaign=AP%C2%AE+Environmental+Science+ +Part+3%3A+Pollution+and+Resources&source=edX&product_category=course&placement_url=https%3A%2F%2Fwww.edx.org%2Flearn%2Fenvironmental-science
- http://ecoursesonline.iasri.res.in/Courses/Environmental%20Science-I/Data%20Files/pdf/lec07.pdf
- https://www.youtube.com/watch?v=5QxxaVfgQ3k

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
2	0	0	2

OPTIMIZATION TECHNIQUES

Course Objectives:

1. To define an objective function and constraint functions in terms of design variables, and then state the optimization problem.
2. To state single variable and multi variable optimization problems, without and with constraints.
3. To explain linear programming technique to an optimization problem, define slack and surplus variables, by using Simplex method.
4. To state transportation and assignment problem as a linear programming problem to determine Simplex method.
5. To study and explain nonlinear programming techniques, unconstrained or constrained, and define exterior and interior penalty functions for optimization problems.

Course Outcomes: At the end of the course, student will be able to

- State and formulate the optimization problem, without and with constraints, by using design variables from an engineering design problem.
- Apply classical optimization techniques to minimize or maximize a multi-variable objective function, without or with constraints, and arrive at an optimal solution.
- Apply and Solve transportation and assignment problem by using Linear programming Simplex method.
- Apply gradient and non-gradient methods to nonlinear optimization problems and use interior or exterior penalty functions for the constraints to derive the optimal solutions
- Formulate and apply Dynamic programming technique to inventory control, production planning, engineering design problems etc. to reach a final optimal solution from the current optimal solution.

UNIT I: Introduction and Classical Optimization Techniques:

Statement of an Optimization problem, design vector, design constraints, constraint surface, objective function, objective function surfaces, classification of Optimization problems.

Classical Optimization Techniques: Single variable Optimization, multi variable Optimization without constraints, necessary and sufficient conditions for minimum/maximum, multivariable Optimization with equality constraints. Solution by method of Lagrange multipliers, multivariable Optimization with inequality constraints, Kuhn – Tucker conditions

UNIT II: Linear Programming :

Standard form of a linear programming problem, geometry of linear programming problems, definitions and theorems, solution of a system of linear simultaneous equations, pivotal reduction of a general system of equations, motivation to the simplex method, simplex algorithm.

UNIT III: Transportation Problem:

Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel's approximation method, testing for optimality of balanced transportation problems,

Special cases in transportation problem.

,

UNIT IV: Nonlinear Programming:

Unconstrained cases, One – dimensional minimization methods: Classification, Fibonacci method, Univariate method, steepest descent method. Constrained cases– Characteristics of a constrained problem, Classification, Basic approach of Penalty Function method, Basic approaches of Interior and Exterior penalty function methods,

UNIT V: Dynamic Programming:

Dynamic programming multistage decision processes, types, concept of sub optimization and the principle of optimality, computational procedure in dynamic programming, examples illustrating the calculus method of solution, examples illustrating the tabular method of solution.

Textbooks:

1. "Engineering optimization: Theory and practice", S. S.Rao, New Age International (P) Limited, 3rd edition, 1998.
2. "Introductory Operations Research", H.S. Kasene& K.D. Kumar, Springer (India), Pvt.LTd.

Reference Books:

1. "Optimization Methods in Operations Research and systems Analysis", by K.V. Mital and C. Mohan, New Age International (P) Limited, Publishers, 3rd edition, 1996.
2. Operations Research, Dr.S.D.Sharma, Kedarnath, Ramnath& Co

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

PROBABILITY & STATISTICS

L	T	P	C
3	0	0	3

Course Objectives:

L	T	P	C
3	0	0	3

The main objectives of this course are:

- To familiarize the students with the foundations of probability and statistical methods
- To impart probability concepts and statistical methods in various applications Engineering

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** classify the concepts of data science and its importance
- **CO 2**: interpret the association of characteristics and through correlation and regression tools
- **CO 3**: apply discrete and continuous probability distributions
- **CO 4:** design the components of a classical hypothesis test
- **CO 5:** infer the statistical inferential methods based on small and large sampling tests

Unit – I:

Descriptive statistics and methods for data science:

Data science – Statistics Introduction – Population vs Sample –Collection of data – primary and secondary data – Type of variable: dependent and independent Categorical and Continuous variables – Data visualization – Measures of Central tendency – Measures of Variability – Skewness – Kurtosis.

UNIT – II:

Correlation and Regression:

Correlation – Correlation coefficient – Rank correlation.

Linear Regression: Straight line – Multiple Linear Regression - Regression coefficients and properties **–** Curvilinear Regression: Parabola – Exponential – Power curves.

UNIT – III:

Probability and Distributions:

Probability– Conditional probability and Baye's theorem – Random variables – Discrete and Continuous random variables – Distribution functions – Probability mass function, Probability density function and Cumulative distribution functions – Mathematical Expectation and Variance – Binomial, Poisson, Uniform and Normal distributions.

UNIT – IV:

Sampling Theory:

Introduction – Population and Samples – Sampling distribution of Means and Variance

(definition only) – Point and Interval estimations – Maximum error of estimate – Central limit theorem (without proof) – Estimation using t, and F-distributions.χ^2

UNIT - V: Tests of Hypothesis:

Introduction – Hypothesis – Null and Alternative Hypothesis – Type I and Type II errors – Level of significance – One tail and two-tail tests – Test of significance for large samples and Small Samples: Single and difference means – Single and two proportions –Student's t- test, F-test, χ^2-test.

Text Books:

1. **Miller and Freund's**, Probability and Statistics for Engineers,7/e, Pearson, 2008.
2. **S. C. Gupta and V.K. Kapoor**, Fundamentals of Mathematical Statistics, 11/e, Sultan Chand & Sons Publications, 2012.

Reference Books:

1. **Shron L. Myers, Keying Ye, Ronald E Walpole,** Probability and Statistics Engineers and the Scientists,8th Edition, Pearson 2007.
2. **Jay l. Devore,** Probability and Statistics for Engineering and the Sciences, 8th Edition, Cengage.
3. **Sheldon M. Ross**, Introduction to probability and statistics Engineers and the Scientists, 4th Edition, Academic Foundation, 2011.
4. **Johannes Ledolter and Robert V. Hogg**, Applied statistics for Engineers and Physical Scientists, 3rd Edition, Pearson, 2010.

Online Resources:

1. https://archive.nptel.ac.in/courses/111/105/111105090/
2. https://onlinecourses.nptel.ac.in/noc21_ma74/preview

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
3	0	0	3

OPERATING SYSTEMS

Course Objectives:
The main objectives of the course are:

- Understand the basic concepts and principles of operating systems, including process management, memory management, file systems, and Protection
- Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
- Illustrate different conditions for deadlock and their possible solutions.

Course Outcomes:
At the end of the course, the student will be able to

- **CO 1:** understand operating system overview, functions and service
- **CO 2:** analyze process concept, inter process communication and synchronization
- **CO 3:** understand memory management and file system interface
- **CO 4:** understand secondary storage management
- **CO 5:** understand protection and security

UNIT - I
Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source Operating Systems **System Structures:** Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Building and Booting an Operating System, Operating system debugging

UNIT - II
Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication.
Threads and Concurrency: Multithreading models, Thread libraries, Threading issues. **CPU Scheduling:** Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling.

UNIT – III
Synchronization Tools: The Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Monitors, Classic problems of Synchronization.
Deadlocks: system Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from Deadlock.

UNIT - IV
Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging, Structure of the Page Table, Swapping.
Virtual Memory Management: Introduction, Demand paging, Copy-on-write, Page replacement, Allocation of frames, Thrashing Storage Management: Overview of Mass Storage Structure, HDD Scheduling.

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

UNIT - V

File System: File System Interface: File concept, Access methods, Directory Structure; File system Implementation: File-system structure, File-system Operations, Directory implementation, Allocation method, Free space management;
File-System Internals: File-System Mounting, Partitions and Mounting, File Sharing.
Protection: Goals of protection, Principles of protection, Protection Rings, Domain of protection, Access matrix.

Text Books:

1. Operating System Concepts, Silberschatz A, Galvin P B, Gagne G, 10th Edition, Wiley, 2018.
2. Modern Operating Systems, Tanenbaum A S, 4th Edition, Pearson, 2016

Reference Books:

1. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018
2. Operating Systems: A Concept Based Approach, D.M Dhamdhere, 3rd Edition, McGraw-Hill, 2013

Online Resources:

1. https://nptel.ac.in/courses/106/106/106106144/
2. http://peterindia.net/OperatingSystems.html

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
3	0	0	3

DATABASE MANAGEMENT SYSTEMS

Course Objectives:
The main objectives of the course are to:

- Introduce database management systems and to give a good formal foundation on the relational model of data and usage of Relational Algebra
- Introduce the concepts of basic SQL as a universal Database language
- Demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- Provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques

Course Outcomes:
At the end of the course, the student will be able to

- **CO 1**: appreciate the underlying concepts of database system architecture and design E-R models to represent simple database application scenarios
- **CO 2**: convert the ER-model to relational tables, populate relational database and formulate SQL queries on data
- **CO 3**: improve the database design by normalization
- **CO 4**: apply and relate the concept of transaction, concurrency control and recovery in database
- **CO 5**: familiar with basic database storage structures and access techniques using Indexing, hashing including B tree methods

UNIT I:
Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; 3 - tier schema architecture for data independence; Database system structure.
Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

UNIT II:
Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus. BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update).

UNIT III:
SQL: Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions (Date and Time, Numeric, String conversion). Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries,

grouping, aggregation, ordering, implementation of different types of joins, view (updatable and non-updatable), relational set operations.

UNIT IV:

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preserving decomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form (BCNF), MVD, Fourth normal form(4NF), Fifth Normal Form (5NF).

UNIT V:

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.

Introduction to Indexing Techniques: B+ Trees, operations on B+Trees, Hash Based Indexing:

Text Books:

1) Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH (For Chapters 2, 3, 4)
2) Database System Concepts,5th edition, Silberschatz, Korth, Sudarsan, TMH (For Chapter 1 and Chapter 5)

Reference Books:

1) Introduction to Database Systems, 8thedition, C J Date, Pearson.
2) Database Management System, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson
3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Online Resources:

1) https://nptel.ac.in/courses/106/105/106105175/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012758066672820_22456_shared/overview

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
3	0	0	3

PYTHON PROGRAMMING

Course Objectives:

The main objectives of the course are:

- To learn about Python programming language syntax, semantics, and the runtime environment
- To be familiarized with universal computer programming concepts like data types, containers
- To be familiarized with general computer programming concepts like conditional execution, loops &functions
- To be familiarized with general coding techniques and object-oriented programming

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** develop essential programming skills in computer programming concepts like data types,containers
- **CO 2:** apply the basics of programming in the Python language
- **CO 3:** solve coding tasks related conditional execution, loops
- **CO 4:** solve coding tasks related to the fundamental notions and techniques used in object-oriented programming.
- **CO 5:** understand and behaviors GUI-Based Programs writing test cases, running tests.

UNIT I

Introduction: Introduction to Python, Program Development Cycle, Input, Processing, and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations, Operators. Type conversions, Expressions, more aboutData Output.

Data Types, and Expression: Strings Assignment, and Comment, Numeric Data Types and Character Sets.

Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables.

Programming**:** Introduction to Programming Concepts with Scratch.

UNIT II

Repetition Structures: Introduction, while loop, for loop, Nested Loops.

Control Statement: Definite iteration for Loop Formatting Text for output, Selection if and if else Statement Conditional Iteration the While Loop

Strings and Text Files: Accessing Character and Substring in Strings, Data Encryption, Strings and Number Systems, String Methods TextFiles, string pattern matching. Understanding read functions, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Manipulating file pointer using seek, Programming using file operations

UNIT III

List and Dictionaries: Lists, tuple, Dictionaries and sets, frozen sets.

Design with Function: Defining Simple Functions, Functions as Abstraction Mechanisms, Problem Solving with Top-Down Design, Design with Recursive Functions, Managing a Program's Namespace, Higher Order Function.

Modules: Modules, Standard Modules, Packages.

UNIT IV

Object Oriented Programming: Concept of class, object and instances, Constructor, class attributes and destructors, Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes, Programming using Oops support.

Design with Classes: Objects and Classes, Data modeling Examples, Case Study An ATM, Structuring Classes with Inheritance and Polymorphism.

UNIT V

Errors and Exceptions: Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Redefined Clean-up Actions.

Graphical User Interfaces: The Behavior of Terminal Based Programs and GUI -Based, Programs, Coding Simple GUI-Based Programs, Other Useful GUIResources, Turtle Graphics

Testing: Basics of testing? unit testing in python, writing test cases, running tests.

Text Books

1. Fundamentals of Python First Programs, Kenneth. A. Lambert, Cengage.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson.
3. Reema Thareja, Python Programming using problem solving Approach, Oxford UniversityPress 2017
4. R. Nageswara Rao core python Programming second Edition.

Reference Books:

1. Introduction to Python Programming, Gowrishankar.S, Veena A, CRC Press.
2. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Online Resources:

1) https://www.tutorialspoint.com/python3/python_tutorial.pdf
2) https://bugs.python.org/file47781/Tutorial_EDIT.pdf

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
0	0	3	1.5

DATABASE MANAGEMENT SYSTEMS LAB

Course Objectives:

The main objectives of the course are to:

- Populate and query a database using SQL DDL/DML Commands
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming PL/SQL including procedures, functions, cursors and triggers

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** design and implement a database schema for given problem.
- **CO 2:** apply the normalization techniques for development of application software to realistic problems.
- **CO 3:** formulate queries using SQL DML/DDL/DCL commands.

Experiments covering the topics:

- DDL, DML, DCL commands
- Queries, nested queries, built-in functions,
- PL/SQL programming- control structures
- Procedures, Functions, Cursors, Triggers,
- Database connectivity- ODBC/JDBC

Sample Experiments:

1. Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2. Queries (along with sub-Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example: - Select the roll number and name of the student who secured fourth rank in the class.
3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5.
   i. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
   ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using

ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.

8. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
11. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers
12. Create a table and perform the search operation on table using indexing and non- indexing techniques.

Text Books/Suggested Reading:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007.

Reference Books:

1) Introduction to Database Systems, 8th edition, C J Date, Pearson.
2) Database Management System, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson
3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
0	0	3	1.5

PYTHON PROGRAMMING LAB

Course Objectives:
The main objectives of the course are:

- To acquire programming skills in core Python.
- To acquire Object Oriented Skills in Python
- To develop the skill of designing Graphical user Interfaces in Python
- To develop the ability to write database applications in Python

Course Outcomes:
At the end of the course, the student will be able to

- **CO 1:** write, Test and Debug Python Programs
- **CO 2:** use conditionals and Loops for Python Programs
- **CO 3:** use functions and represent Compound data using Lists, Tuples and Dictionaries
- **CO 4:** use various applications using python
- **CO 5:** use GUI application to develop user interfaces

List of Experiments:

1) Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.

2) Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average.

3) Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86,89.

4) Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.

5) Use a for loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.

```
*
**
***
****
```

6) Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.

7) Write a program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and not close otherwise.

8) Write a program that asks the user to enter a word and prints out whether that word contains any vowels.

9) Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe

10) Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be1,000,000.

11Write a program that generates a list of 20 random numbers between 1 and100.
(a)Print the list.
(b)Print the average of the elements in the list.
(c)Print the largest and smallest values in the list.
(d)Print the second largest and second smallest entries in the list
(e)Print how many even numbers are in the list.

12) Write a program to use split and join methods in the given string and store them in a dictionary data structure.
13) Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0].

14) Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.

15) Write a function called sum_digitsthat is given an integer num and returns the sum of the digits of num.

16) Write a function called first_diffthat is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return-1.

17) Write a function called number_of_factorsthat takes an integer and returns how many factors the number has.

18) Write a function called is_sortedthat is given a list and returns True if the list is sorted and False otherwise.

19) Write a function called root that is given a number x and an integer n and returns x1/n. In the function definition, set the default value of n to 2.

20) Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sortedlist.
(a) Do this using the sort method.
(b) Do this without using the sort method

21) Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user's word.

22) Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons.

23) Write a program that reads a list of temperatures from a file called temps.txt, converts those temperatures to Fahrenheit, and writes the results to a file calledftemps.txt.

24) Write a class called Product. The class should have fields called name, amount, and price, holding the product's name, the number of items of that product in stock, and the regular price of the product. There should be a method get_pricethat receives the number of items to be bought and returns the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called make_purchasethat receives the number of items to be bought and decreases amount by that much.

25) Write a class called Time whose only field is a time in seconds. It should have a method called convert_to_minutesthat returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called convert_to_hoursthat returns a string of hours, minutes, and seconds formatted analogously to the previous method.

26) Write a Python class to implement pow(x,n).

27) Write a Python class to reverse a string word byword.

28Write a program to demonstrate Try/except/else.

29) Write a function nearly _equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

30) Write a python program to create wheel using turtle graphics.

31)Write a python program on GUI to create a registration form.

32)Write a python program to check whether a string starts and ends with the same character or not (using Regular Expression re module).

Text Books:

1)Fundamentals of Python First Programs, Kenneth. A. Lambert, Cengage.
2)Python Programming: A Modern Approach, Vamsi Kurama, Pearson.
3) Reema Thareja, Python Programming using problem solving Approach, Oxford University Press 2017
4) R. Nageswara Rao core python Programming second Edition.

Reference Books:

1)Introduction to Python Programming, Gowrishankar.S, Veena A, CRC Press.
2)Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Online Resources:

1) https://www.tutorialspoint.com/python3/python_tutorial.pdf
2) https://bugs.python.org/file47781/Tutorial_EDIT.pdf

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
0	1	2	2

FULL STACK DEVELOPMENT (SKILL ENHANCEMENT COURSE-2)
(Bootstrap, REACT JS, NODE JS)

Course Objectives:

The main objectives of the course are:

- To implement the static web pages using HTML and do client-side validation using JavaScript.
- To design and work with databases using Java.
- To develop an end-to-end application using java full stack.
- To introduce Node JS implementation for server-side programming.
- To experiment with single page application development using React.

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** build a custom website with HTML, CSS, and Bootstrap and little JavaScript.
- **CO 2:** demonstrate Advanced features of JavaScript and learn about JDBC.
- **CO 3:** develop Server – side implementation using Java technologies like.
- **CO 4:** develop the server – side implementation using Node JS.
- **CO 5:** design a Single Page Application using React.

List of Experiments:

[HTML, CSS, Bootstrap, Java Script]

1. Build a responsive web application for shopping cart with registration, login, catalog and cart pages using CSS3 features, flex and grid.
2. Make the above web application responsive web application using Bootstrap framework.
3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1 and experiment 2.

[React JS]

4. Install ReactJS and Environment Setup.
5. Creating a simple ReactJS Application.
6. Event Handling in ReactJS.
7. Use Props in React JS to Customize React Components.
8. Passing Data Between React JS Components Using Props.
9. Create a TODO application in react with necessary components and deploy it into github.
10. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js
11. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

[Node JS]

12. Install NodeJS and Environment Setup.
13. Create a calculator Module
14. File operations using nodeJS
15. Connect to MySQL database and perform database operations.
16. Parse URL parameters, Parse a JSON file.
17. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2 nd Edition, A Press.

Online Resources:

1. https://www.almabetter.com/bytes/tutorials/reactjs/higher-order-components
2. https://www.tutorialkart.com/nodejs/node-js-examples/#gsc.tab=0

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
B. Tech in Information Technology

II Year II Semester

L	T	P	C
0	0	2	1

DESIGN THINKING & INNOVATION

Course Objectives:

The objectives of the course are to:

- Bring awareness on innovative design and new product development.
- Explain the basics of design thinking.
- Familiarize the role of reverse engineering in product development.
- Train how to identify the needs of society and convert into demand.
- Introduce product planning and product development process.

Course Outcomes:

At the end of the course, the student will be able to

- **CO 1:** define the concepts related to design thinking.
- **CO 2:** explain the fundamentals of Design Thinking and innovation.
- **CO 3:** apply the design thinking techniques for solving problems in various sectors.
- **CO 4**: analyze to work in a multidisciplinary environment.
- **CO 5:** evaluate the value of creativity.

UNIT – I

Introduction to Design Thinking

Introduction to elements and principles of Design, basics of design-dot, line, shape, form as fundamental design components. Principles of design. Introduction to design thinking, history of Design Thinking, New materials in Industry.

UNIT - II

Design Thinking Process

Design thinking process (empathize, analyze, idea & prototype), implementing the process in driving inventions, design thinking in social innovations. Tools of design thinking - person, costumer, journey map, brainstorming, product development

Activity: Every student presents their idea in three minutes, every student can present design process in the form of flow diagram or flow chart etc. Every student should explain about product development.

UNIT - III

Innovation

Art of innovation, Difference between innovation and creativity, role of creativity and innovation in organizations. Creativity to Innovation. Teams for innovation, Measuring the impact and value of creativity.

Activity: Debate on innovation and creativity, Flow and planning from idea to innovation, Debate on value-based innovation.

UNIT - IV

Product Design

Problem formation, introduction to product design, Product strategies, Product value, Product planning, product specifications. Innovation towards product design Case studies.

Activity: Importance of modeling, how to set specifications, Explaining their own product design.

UNIT – V

Design Thinking in Business Processes

Design Thinking applied in Business & Strategic Innovation, Design Thinking principles that redefine business – Business challenges: Growth, Predictability, Change, Maintaining Relevance, Extreme competition, Standardization. Design thinking to meet corporate needs. Design thinking for Startups. Defining and testing Business Models and Business Cases. Developing & testing prototypes.

Activity: How to market our own product, about maintenance, Reliability and plan for startup.

Textbooks:

1. Tim Brown, Change by design, 1/e, Harper Bollins, 2009.
2. Idris Mootee, Design Thinking for Strategic Innovation, 1/e, Adams Media, 2014.

Reference Books:

1. David Lee, Design Thinking in the Classroom, Ulysses press, 2018.
2. Shrrutin N Shetty, Design the Future, 1/e, Norton Press, 2018.
3. William lidwell, Kritinaholden, &Jill butter, Universal principles of design, 2/e, Rockport Publishers, 2010.
4. Chesbrough.H, The era of open innovation, 2003.

Online Resources:

- https://nptel.ac.in/courses/110/106/110106124/
- https://nptel.ac.in/courses/109/104/109104109/
- https://swayam.gov.in/nd1_noc19_mg60/preview
- https://onlinecourses.nptel.ac.in/noc22_de16/preview