School of Engineering & Technology

B.Tech. –Computer Science and Engineering

Semester III Course Structure & Syllabus

Theory/ Practical	Sl. No	Course Type	Course Code	Course Name	Hours per week			Credit
					L	T	P	C
T	1.	ESC	BCS23201T	Discrete Mathematics	3	1	0	4
T	2.	PCC	BCS23202T	Object Oriented Programming using C++	3	0	0	3
P	3	PCC	BCS23202P	Object Oriented Programming using C++ LAB	0	0	4	2
T	6	PCC	BCS23203T	Data Structure and Algorithms	3	0	0	3
P	7	PCC	BCS23203P	Data Structure and Algorithms Lab	0	0	4	2
T	4	PCC	BCS23204T	Python Programming	3	0	0	3
P	5	PCC	BCS23204P	Python Programming Lab	0	0	2	1
T	8	ESC	BCS23205T	Digital Electronics	3	0	0	3
T	9	HSMC	BCH23112T	Environmental Science	2	0	0	0
				Total	**17**	**1**	**10**	**21**

Discrete Mathematics	L	T	P	C
	4	0	0	4

Pre-requisite:

1. Basic of Mathematics.

Course Objectives:

1. To introduce the concept of set theory.
2. To introduce about the vector spaces
3. Perform basic matrix operations including sums, products, and transpose etc.

Course Outcome:

After successful completion of the course, the students will learn

CO1: Solve and analyses real world engineering problems by applying set theory, relations, construct and use functions and apply these concepts to solve problems

CO2: Be able to apply the fundamental concepts of Partial differential Equations.

CO3: Analyze and apply the concepts of Matrices and propositional logic to solve problems and to construct proofs using mathematical induction

CO4: be able to construct simple mathematical proofs and possess the ability to analyze them.

MODULE 1: Set, Relation and Function — **10 hours**

Sets, relations, properties of binary relations, closures of relation, equivalence relations, equivalence classes and partitions. Partial ordering relations.

MODULE 2: Matrices — **10 hours**

Row and column operations, vectors and matrices, partitioning of matrices, representing relations using matrices, Determinant of a square matrix, minor, cofactor, the Cayley Hamilton theorem, inverse of a matrix, product form of inverse. Rank of a matrix. Solutions of simultaneous linear equations, existence of solutions, solution by Gaussian elimination, Eigen values and Eigen vectors.

MODULE 3: Number Theory — **10 hours**

Basic of counting principles, principle of inclusion exclusion, application of inclusion and exclusion. Pigeonhole principle, generalized Pigeonhole principle and its application, permutations and combinations, permutations with repetitions, combinations with repetitions, permutations of sets with indistinguishable objects.

MODULE 4:Partial Differential Equation — **10 hours**

First order Partial differential equation: Partial differential equation of first order, Linear partial differential equation, Non-linear partial differential equation Homogenous and non-homogeneous partial differential equation with constant coefficient

MODULE 5: Vector Spaces — **10 hours**

Definition and examples of vector spaces. Elementary properties of R as a vector space. Subspaces of a vector space. Union, intersection and sum of two subspaces. Subspaces generated by a subset of a vector space. Definition, example and properties of linearly independent and dependent set of vectors. Basis and dimension of a vector space. Examples of finite dimensional

vector spaces.	
MODULE 6: Logic	**10 hours**
Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers.	
Total Lecture hours	**60 hours**
Text Book/Reference Book:	
1. E. Kreyszig," Advanced Engineering Mathematics:, Eighth Edition, Wiley India.	
2. B.V. Ramana, "Higher Engineering Mathematics", McGraw Hil Education.	
3. N.P.Bali and Manish Goel, " A text book of Engineering mathematics", Laxmi Publication.	
4. B. S. Grewal ,"Higher Engineering Mathematics", Khanna Publication, Delhi.	
5. Discrete Mathematics and Its Applications, by Kenneth H. Rosen, Tata McGraw Hill, 6th edition, ISBN: 0072880082© 2007.	
6. Elements of Discrete Mathematics, by C. L. Liu, Tata McGraw Hill Education Private Limited, 3rd edition,2008	
7. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press, Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson	
8. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and Its Application to Computer Science", TMG Edition, Tata McGraw-Hill	

	OBJECT ORIENTED PROGRAMMING USING C++	L	T	P	C
		3	0	0	3

Pre-requisite:

1. Basic of computer knowledge.
2. Knowledge of C programming.

Course Objectives:

1. To introduce the concept of object orientation to C++.
2. To introduce solving real world problems.
3. It will help in acquainting the techniques and applications of C++ for programming based onchallenging tasks.

Course Outcome:

After successful completion of the course, the students will learn

CO1: Develop the concept of object-oriented programming, its applications, and its differences with procedure-oriented programming.
CO2: Implement the concept of class, object, and constructor in writing codes for solving problems.
CO3: Apply the concept of polymorphism, inheritance in solving engineering problems.
CO4: Design programmed solution using the concept of files and pointers including templates, exceptions and file handling.

.

MODULE 1: Introduction to Object Oriented Programming — **5 hours**

Introduction to Object Oriented Programming: Computer programming background- C++ overview. First C++ Program -Basic C++ syntax, Object Oriented Programming: Concept of object, Classes, methods and messages, abstraction and encapsulation, inheritance, abstract classes, polymorphism.

MODULE 2: Basics of C++ Programming — **6 hours**

C++ Program Structure, Character Set and Tokens, Data Type, Type Conversion, Preprocessor Directives, Namespace, Input/output Streams, Control Statements.
Keywords, Identifiers and constants, Operators in C++, Scope resolution operator. Function: Function Overloading, Inline Function, Default Argument, Expressions, Call by Value, Call by reference – Return by reference.
Pointers: Pointer variable declaration & initialization, Operators in pointers, Pointers and Arrays, Pointer and Function.

MODULE 3: Class and Objects — **12 hours**

A Simple Class and Object, Accessing members of class, Initialization of class objects (Constructor, Destructor), Default Constructor, Parameterized Constructor, Copy Constructor , Default Constructor, Objects as Function Arguments, Returning Objects from Functions, Memory allocation for Objects, Static members, Member functions defined outside the class.

MODULE 4:Operator Overloading	**3 hours**
Fundamentals of operator overloading, Restriction on operator overloading, Operator function as a class member, Overloading unary and binary operator.	
MODULE 5: Inheritance	**10 hours**
Introduction to inheritance, Derived Class and Base Class, Access Specifiers (private, protected, and public), Types of inheritance, Constructor and Destructor in derived class, Aggregation.	
MODULE 6: Polymorphism, Virtual Function and miscellaneous C++ features	**4hours**
Concept of Virtual functions, Late Binding, Abstract class and pure virtual functions, Virtual Destructor, Virtual Base class, Friend function and static function, inline function, This pointer, Concrete classes, Polymorphism and its roles.	
MODULE 7: File handling, Function Templates and Exception Handling	**5 hours**
Stream Class Hierarchy for Console Input/output, Function template, Function template with multiple arguments, Class template, Exception Handling (Try, throw and catch), Use of exception handling.	
Total Lecture hours	**45 hours**

Text Book:

1.E Balaguruswamy (2013), Object-oriented programming with C++, 6th Edition,
Mc GrawHill Education.

2. Bjarne Stroustrup (2013), The C++ Programming Language, 4th Edition, Addison-Wesly.

3. Herbert Schildt (2017), C++: The Complete Reference, 4th Edition, McGraw Hill Education.

Reference Books:

1. Reema Thareja,(2016),Object Oriented Programming with C++,1st Edition, Oxford University.

OBJECT ORIENTED PROGRAMMING USING C++ LABORATORY		
List of Lab Experiments		
Lab 1	Basics of C++ Programming.	**2hours**
Lab 2	WAP using Switch Case to add, subtract, multiply and divide of two numbers.	**1 hour**
Lab 3	WAP to illustrate Class Declaration, Definition, Member function, objects. (Area of Trapezium, Rhombus, Circle, Triangle).	**4 hours**
Lab 4	WAP to create a simple class named Account and write methods to deposit and withdraw amount from the account.	**2 hours**
Lab 5	WAP to demonstrate the usage of a Constructor and Destructor in a class.	**1 hour**
Lab 6	WAP to illustrate parameterized constructor.(default, copy constructor)	**2 hours**
Lab 7	WAP to demonstrate: a) Operator Overloading b) Function Overloading.	**2 hours**
Lab 8	WAP to demonstrate Hybrid Inheritance. (Single, Multiple, Multi-level, Hierarchical)	**5 hours**
Lab 9	WAP to demonstrate Friend Function and Friend class.	**2 hours**
Lab 10	WAP to demonstrate polymorphism by calculating area of a rectangle and triangle using Shape class.	**2 hours**
Lab 11	WAP to demonstrate Virtual function.	**2 hours**
Lab 12	WAP to overload +operator to add two numbers.	**1hour**
Lab 13	WAP to create a Class Template.	**2 hours**
Lab 14	WAP to demonstrate exception handling.	**2 hours**
	Total	**30hours**

Text Book:

(4) E Balaguruswamy (2013), Object-oriented programming with C++, 6th Edition, McGrawHill Education.

(5) Bjarne Stroustrup (2013), The C++ Programming Language, 4th Edition, Addison-Wesly.

(6) Herbert Schildt (2017), C++: The Complete Reference, 4th Edition, McGraw Hill Education.

Reference Books

(2) Reema Thareja,(2016),Object Oriented Programming with C++,1st Edition, Oxford University.

Course Title	Hours per week L-T-P	Credit C
Data Structure and Algorithms	**3-0-4**	**5**

Course Outcome (Theory)

Course Outcome	Statement
CO1	Analyze the performance of various algorithms
CO2	Make use of the functionality of linear data structure
CO3	Model the solutions using non-linear data structure
CO4	Choose appropriate searching and sorting technique for a given problem
CO5	Implementation of graphs

MODULE 1: (3 Lectures)

Notion of datastructures and algorithms, understanding growth of functions, Worst-case, average case and best case time/space complexity, Asymptotic Notation.

MODULE 2: (8 Lectures)

Abstract data-type (ADTs): arrays and linked list ADTs, Stacks, Queues: ADTs and implementations using arrays, linked lists, Doubly linked lists: ADT and implementation, Dictionary ADT: implementation using array, linked lists, binary search, Tree ADT and examples, Implementation of trees and basic traversal algorithms.

MODULE 3: (3 Lectures)

Priority Queue ADT, Definition of heaps, Implementation of Priority Queues using heaps and running time analysis, Implementation of heaps using arrays, Heap-sort.

MODULE 4: (9 Lectures)

Binary Search Trees: definition and some basic algorithms, Implementation of Dictionary ADTs using Binary Search trees and running time analysis, AVL trees: height balance condition, rotations, and implementation of dictionary ADT, 2-4 Trees: Multi-way search trees, implementation of dictionary ADT, Informal discussion of extension to B-trees.

MODULE 5: (6 Lectures)

Map ADT, Hash Tables and implementation of Map using Hash Tables, Design of hash functions, Collision resolution schemes: chaining, open addressing schemes like linear probing, quadratic probing, double hashing, Applications of Hashing: finding duplicates, set

intersection, etc, Tries: implementation of Map ADT using tries, Compressed tries and suffix tries.

MODULE 6: (4 Lectures)

Bubble sort, insertion sort, selection sort, Merge sort and divide and conquer paradigm, Quick sort: average and worst case analysis, randomized quicksort, Selection based on partitioning ideas used in Quick Sort.

MODULE 7: (8 Lectures)

Graph ADTs and applications, Adjacency list and adjacency matrix representations and relative merits, Basic graph definitions: paths, cycles, trees, spanning trees, connected components, Euler's formula, Depth First Search Traversal algorithm for directed graphs: classification of edges into forward, back and cross edges. Applications to cycle finding, topological sort in directed acyclic graphs, finding connected components. Running time analysis, Breadth first search algorithm: implementation using queues, shortest path tree property. Running time analysis.

Text Books/ Reference Books:

1."Data Structures and Algorithms in Java", by Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons; 3rd Edition.
2."Data Structures and Algorithms in Python", by Michael T. Goodrich and Robert, Tamassia, Wiley, 1st Edition.
3. In case any other programming language is used for this course, some other suitable text book may be chosen

GIRIJANANDACHOWDHURYUNIVERSITY
Hathkhowapara,Azara,Guwahati781017,Assam

BSC23203P	DATA STRUCTURE AND ALGORITHMS	L	T	P	C
		0	0	4	2

LIST OF EXPERIMENTS

1. Write a program to implement bubble sort, insertion sort and selection sort in a menu driven program.
2. Write a program to perform linear search and binary serch.
3. Write a program to perform operations in an array.
4. Write a program to implement stack using array.
5. Write a program to implement queue using array.
6. Write a program to implement circular queue, priority queue.
7. Write a program to implement singly linked list along with operations.
8. Write a program to implement circular doubly linked list along with operations.
9. Write a program to create a binary search tree with operations of searching and deletion.
10. Write a program to perform traversal of a binary search tree.
11. Write a program to represent graph in memory and perform breadth first search and depth first search on this graph.

Python Programming	L	T	P	C
	3	0	0	3

Pre-requisite: Basic knowledge of Programming

Course Objectives:

1. To be able to introduce core programming basics and program design with functions using Python programming language.
2. To understand a range of in-built functions as well as in-depth data and information processing techniques.

Course Outcome:

After successful completion of the course, the students will be able to

CO1: Interpret the fundamental Python syntax and semantics and determine the methods to create and manipulate Python programs by utilizing the data structures like lists, dictionaries, tuples and sets.
CO 2: Express proficiency in the handling of control flow statements, strings and functions.
CO 3: Identify the commonly used operations involving file systems and regular expressions.
CO 4: Evaluate the python in built functions, Scientific computing packages
CO 5: Evaluate the various feature engineering algorithms by python programming.

Module 1: Introduction to Programming in Python	**10 hours**

Introduction to Programming in Python: What Is Python? Features of Python, Python environment set up: Installing Python, Running Python, Python Documentation, Structure of a Python Program Basics of Programming in Python: Input statement, output statement, variables, operators, numbers, Literals, strings, lists and tuples, dictionaries.

Module 2: Conditionals, Loops and Functions.	**12 hours**

Conditionals and Loops: if statement, else Statement, elif Statement, while Statement, for Statement break Statement, continue Statement, pass Statement. Functions: Built-in Functions, User defined functions: Defining a Function, Calling a Function, Various Function Arguments.

Module 3: Files, Modules and Introduction to Advanced Python.	**10 hours**

Files: File Objects, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Scientific computing packages: **Numpy:** -Array object, creating array, matrix, indexing, slicing, resizing, reshaping, arithmetic operations, functions, matrices and vector operations, **Matplotlib**: basic plotting, **Scipy:** Linear algebra operations, equation solving. Regular Expressions.

Module 4: Introduction to advance Python Programming	**7 hours**

OOPS

Module 5: Python GUI & CGI Programming and Python database connectivity.	**6 hours**

Python GUI Programming (Tkinter): Tkinter Programming example, Tkinter widges, standard attributes, geometry management

Books

1. Kamthane, A. N., & Kamthane, A.A. Programming and Problem Solving with Python, McGraw Hill Education. 2017.
2. Balaguruswamy E., "Introduction to Computing and Problem Solving using Python",2nd Edition, McGraw Hill Education, 2018.
3. Taneja, S., Kumar, N. "Python Programming- A modular Approach", Pearson Education India, 2018.

Additional References

(i) Mark Lutz, "Learning Python" O'Reilly Media, 2013.

(ii) Guttag, J. V. Introduction to computation and programming using Python. MIT Press. 2018

(iii) Downey, A. B. Think Python–How to think like a Computer Scientist 2nd Edition. O'Reilly 2015

(iv) Robert Johansson, "Numerical Python: Scientific Computing and Data Science .Applications with Numpy, SciPy and Matplotlib" Apress, 2019.

Python Programming LAB	L	T	P	C
	0	0	2	1

Pre-requisite: Basic knowledge of Programming

Course Outcome:

After successful completion of the course, the students will be able to
CO1: Write, Test and Debug Python Programs.
CO 2: Implement Conditionals and Loops for Python Programs
CO 3: Use functions and represent Compound data using Lists, Tuples and Dictionaries.
CO 4: Read and write data from & to files in Python and develop Application

Suggested Practical List	**15 hours**

1. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:
   Grade A : if Percentage >=80
   Grade B : if Percentage >=60 and Percentage <80
   Grade C : if Percentage >=40 and Percentage <60
   Grade D : if Percentage <=40
2. WAP to print factors of a given number.
3. WAP to add N natural numbers and display their sum.
4. WAP to print the following conversion table (use looping constructs):

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

   5.8ft 69.6inches
   5.9ft 70.8inches
   6.0ft 72inches

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

   *

   * *

   * * *

   * * * *
   (for n =4)

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.

7. Write a function that calculates factorial of a number n.

8. WAP to print the series and its sum: (use functions)
   1/1! + 1/2! + 1/3! …….1/n!
9. WAP to perform the following operations on an input string

   a. Print length of the string

   b. Find frequency of a character in the string

   c. Print whether characters are in uppercase or lowercase

10. WAP to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.
11. WAP to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.
12. WAP to create a tuple t1 = (1,2,5,7,2,4). The program should perform the following:
    a. Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.
    b. Concatenate tuple t2 = (10,11) with t1.

	DIGITAL ELECTRONICS	L	T	P	C
		3	0	0	3

Pre-requisite: Basics of logic circuits, Boolean algebra

Course Objectives:

1. To provide the fundamental concepts associated with the digital logic and circuit design.
2. To introduce the basic concepts and laws involved in the Boolean algebra and logic families and digital circuits.
3. To familiarize with the different number systems, logic gates, and combinational and sequential circuits utilized in the different digital circuits and systems.

Course Outcome:

After successful completion of the course, the students will learn

1. CO1: To solve the Boolean algebra, logic gates, logical variables, the truth table, number systems, codes, and their conversion from to others.
2. CO4: To analyse various types of analog to digital converter, memory elements and programmable logic arrays and properties of Digital logic families and programmable logic devices.
3. CO2: To design the most simplified combinational circuit using mapping techniques.
4. CO3: To design different types of sequential circuits using flip flops and counters.

MODULE 1: Introduction to number systems and Boolean Algebra — **10 hours**

Data and number system: Binary, Octal and Hexadecimal representations and their conversion, BCD, ASCII, EBDIC, Gray codes, code conversion, Error detection and correction codes - parity check codes and Hamming code. Representation of Signed binary numbers with 1's and 2's complement methods, Binary arithmetic, Logic circuits, integrated circuits;

Axiomatic definitions of Boolean Algebra - Basic Theorems and Properties of Boolean Algebra, Boolean Functions- Canonical and Standard forms ,Digital Logic Gates, Simplification of Boolean Expressions: The map method- SOP and POS ,NAND and NOR implementation ,Don't Cares - The Tabulation Method – Determination and Selection of Prime Implicants.

Module 2: Combinational logic circuits — **8hours**

Basic logic operation and logic gates, Combinational logic circuit design using truth-table, Different Adders and Subtractors, Comparator, Encoder, Decoder, Multiplexer, Demultiplexer, BCD arithmetic, carry look ahead adder, serial adder, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization. Synthesis of combinational logic circuits.

Module 3: Sequential Logic systems	**12 hours**
Basic sequential circuits- latches and flip-flops: SR-latch, D-latch, D flip-flop, T flip-flop, JK flip-flop; race around condition, master slave conversion of flip-flops, Edge triggered FF, applications of flip flops, Sequential shift register, sequence generator, serial to parallel converter, parallel to serial converter, ring counter, ripple(Asynchronous) counters, synchronous counters, counters design using flip flop, applications of counters.	
Module 4: Logic Families and Semiconductor Memories	**12hours**
Brief idea about DTL, TTL, ECL, MOS and CMOS families and their comparison based on Parameters: fan-in, fan-out, propagation delay, speed-power product, etc. TTL NAND gate, Tristate TTL, ECL, CMOS families and their interfacing. Memory elements, Read-only memory, read/write memory - SRAM and DRAM Concept of Programmable logic devices like PLAs, PALs and their applications, Introduction to field programmable gate arrays (FPGAs), analog to digital converter: quantization and encoding, different types of conversion, accuracy and resolution.	
Total Lecture hours	**42 hours**

Text Book

1.R.P. Jain, “Modern digital Electronics”, Tata McGraw Hill, 4th edition, 2009.

2.S.Salivahanan and S.Arivazhagan, DIGITAL CIRCUITS AND DESIGN, 5th edition,2018

3. Fundamental of digital circuits by A. ANAND KUMAR, PHI Publication.

Reference Books

1.D. P. Kothari and J. S Dhillon, ―Digital Circuits and Design‖, Pearson, 2016,

2. Morris Mano, ―Digital Design‖, Prentice Hall of India, Third Edition.

3..Charles H Roth Jr., Larry L. Kinney ―Fundamentals of Logic Design, Cengage Learning, 7th Edition

4.Digital Fundamentals by FLOYD & JAIN, Pearsons Publication

5.Fundamentals of Logic Design by Charles H. Roth Thomson

Internal Member/BoS External Member/BoS

BCH23112T	ENVIRONMENTAL SCIENCE	L	T	P	C
		2	0	0	0

Pre-requisite: Biology, Sociology, Chemistry

Course Objectives: The objectives of this course are to:

1. To examine the concept of environment and ecosystem.
2. To understand the different types of natural resources and the concept of biodiversity and its importance for the environment.
3. To examine the concept of different types of environmental problems such as pollution, climate change, population growth and its causes, effects on environment and to find out the solution to control the environmental degradation.

Course Outcome: After successful completion of this course, the students will be able to

CO1 : Understand the basic concept of environment and ecosystem.
CO2: Value the natural resources, conservation of biodiversity and its importance.
CO3: Evaluate the problems of environmental issues such as pollution, population growth, climate change and its impact on human and environment and the control measures.

Module1: Concepts of Environmental Science	**3 hours**

Definition of environment, scope and importance of environmental studies; Need for public awareness; Structure and functions in an ecosystem.

Module 2: Natural Resources	**6 hours**

Renewable and Non-renewable Resources;
Forest, water, minerals, food and land resources (with example of one case study); Energy, growing energy needs, energy sources (conventional and alternative).

Module 3: Biodiversity And Its Conservation	**5 hours**

Biodiversity at global, national and local levels; India as a mega diversity nation;
Threats to biodiversity (biotic, abiotic stresses), and strategies for conservation.

Module 4: Environmental Pollution	**8 hours**

Types of pollution-Air, water (including urban, rural, marine), soil, noise, thermal, nuclear; Pollution prevention; Management of pollution –Rural /Urban/Industrial waste management [with case study of any one type, e.g., power (thermal/nuclear), fertilizer, tannin, leather, chemical, sugar], Solid/Liquid waste management, disaster management.

Module 5: Social Issues and Environment	**8 hours**

From unsustainable to sustainable development; Problems relating to urban environment-Population pressure, water scarcity, industrialization, remedial measures;
Climate change-Reasons, effects (global warming, ozone layer depletion, acid rain) with one case study; Legal Issues-Environmental legislation (Acts and issues involved), Environmental ethics.

Total Lecture hours	**30 hours**

Text Book(s)

1	S.C. Santra: Environmental Science, New Central Book Agency
2	S.E. Manahan: Environmental Chemistry
3	K.V. Krishnamurthy: Textbook of Biodiversity

Reference Book(s)

1	Agarwal, K.C., Environmental Biology, Nidi Publication Ltd., Bikaner, 2001.
2	Bharucha Erach, Biodiversity of India, Mapin Publishing Pvt. Ltd., Ahmadabad, 2002
3	Dr R J Ranjit Daniels and Dr Jagadish Krishnaswamy, Environmental studies-2010-Willey India