BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES

COURSE HANDOUT

Part A: Content Design

Course Title	Data Structures and Algorithms Design
Course No(s)	SS ZG519
Credit Units	5
Course Author	A. Baskar
Version No	
Date	

Course Objectives

CO1	Introduce mathematical and experimental techniques to analyze algorithms
CO2	Introduce linear and non-linear data structures and best practices to choose appropriate data structure for a given application
C03	Teach various dictionary data structures (Lists, Trees, Heaps) with illustrations on possible representation, various operations and their efficiency
C04	Exposes students to various sorting and searching techniques
CO5	Discuss in detail various algorithm design approaches (Greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound) with appropriate examples, methods to make correct design choice and the efficiency concerns.
CO6	Introduce complexity classes , notion of NP-Completeness, ways of classifying problem into appropriate complexity class
C07	Introduce reduction method to prove a problem's complexity class.

Text Book(s)

T1	Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition)

Reference Book(s) & other resources

R1	Data Structures, Algorithms and Applications in C++, Sartaj Sahni, Second Ed, 2005, Universities Press
R2	Introduction to Algorithms, TH Cormen, CE Leiserson, RL Rivest, C Stein, Third Ed, 2009, PHI

Content Structure

1. Analyzing Algorithms [3 Hours]
   - 1.1. Theoretical Foundation
     - 1.1.1. Algorithms and it's Specification
     - 1.1.2. Random Access Machine Model
     - 1.1.3. Counting Primitive Operations
     - 1.1.4. Notion of best case, average case and worst case
   - 1.2. Characterizing Run Time
     - 1.2.1. Use of asymptotic notation
     - 1.2.2. Big-Oh Notation, Little-Oh, Omega and Theta Notations
   - 1.3. Correctness of Algorithms
   - 1.4. Analyzing Recursive Algorithms
     - 1.4.1. Recurrence relations
     - 1.4.2. Specifying runtime of recursive algorithms
     - 1.4.3. Solving recurrence equations
   - 1.5. Case Study: Analyzing Algorithms

2. Elementary Data Structures [2 hours]
   - 2.1. Stacks
     - 2.1.1. Stack ADT and Implementation
     - 2.1.2. Applications
   - 2.2. Queues
     - 2.2.1. Queue ADT and Implementation
     - 2.2.2. Applications
   - 2.3. List
     - 2.3.1. Notion of position in lists
     - 2.3.2. List ADT and Implementation

3. Non-Linear Data Structures [4 hours]
   - 3.1. Trees
     - 3.1.1. Terms and Definition
     - 3.1.2. Tree ADT
     - 3.1.3. Applications
   - 3.2. Binary Trees
     - 3.2.1. Properties
     - 3.2.2. Representations (Vector Based and Linked)
     - 3.2.3. Binary Tree traversal (In Order, Pre Order, Post Order)
     - 3.2.4. Applications
   - 3.3. Heaps
     - 3.3.1. Definition and Properties
     - 3.3.2. Representations (Vector Based and Linked)
     - 3.3.3. Insertion and deletion of elements
     - 3.3.4. Heap implementation of priority queue
     - 3.3.5. Heap sort

4. Dictionaries [as Hash Tables and Search Trees] [3 hours]
   - 4.1. Unordered Dictionary
     - 4.1.1. ADT Specification
     - 4.1.2. Applications
   - 4.2. Hash Tables
     - 4.2.1. Notion of Hashing and Collision (with a simple vector based hash table)
     - 4.2.2. Hash Functions
       - 4.2.2.1. Properties
       - 4.2.2.2. Simple hash functions
     - 4.2.3. Methods for Collision Handling

- 4.2.3.1. Separate Chaining
- 4.2.3.2. Notion of Load Factor
- 4.2.3.3. Rehashing
- 4.2.3.4. Open Addressing [Linear & Quadratic Probing, Double Hash]

4.3. Ordered Dictionary
- 4.3.1. ADT Specification
- 4.3.2. Applications

4.4. Binary Search Tree
- 4.4.1. Motivation with the task of Searching and Binary Search Algorithm
- 4.4.2. Properties of BST
- 4.4.3. Searching an element in BST
- 4.4.4. Insertion and Removal of Elements
- 4.4.5. Performance

5. Algorithm Design Techniques [6 Hours]

5.1. Greedy Method
- 5.1.1. Design Principles and Strategy
- 5.1.2. Fractional Knapsack Problem
- 5.1.3. Task Scheduling Problem

5.2. Divide and Conquer
- 5.2.1. Design Principles and Straegy
- 5.2.2. Analyzing Divide and Conquer Algorithms
- 5.2.3. Integer Multiplication Problem
- 5.2.4. Sorting Problem
  - 5.2.4.1. Merge Sort Algorithm
  - 5.2.4.2. Quick Sort Algorithm
- 5.2.5. Searching Problem and Binary Search Algorithm [Ref to 4.4.1]

5.3. Dynamic Programming
- 5.3.1. Design Principles and Strategy
- 5.3.2. Matrix Chain Product Problem
- 5.3.3. 0/1 Knapsack Problem

5.4. Graph Algorithms
- 5.4.1. Introduction to Graphs
- 5.4.2. Prim's and Kruskal's Algorithms [Greedy]
- 5.4.3. Dijkstra’s Algorithm [Greedy]
- 5.4.4. Bellman-ford Shortest Path Algorithm [Greedy]
- 5.4.5. All Pair Shortest Path Algorithm [Dynamic Programming]

6. Complexity Classes [4 hours]

6.1. Definition of P and NP classes and examples

6.2. Understanding NP-Completeness
- 6.2.1. NP-Hardness
- 6.2.2. Polynomial time reducibility
- 6.2.3. Cook-Levin theorem
- 6.2.4. Problems in NP-Complete and using polynomial time reductions
  - 6.2.4.1. CNF-SAT, 3-SAT
  - 6.2.4.2. Vertex Cover
  - 6.2.4.3. Clique and Set-Cover
  - 6.2.4.4. Subset-Sum and Knapsack
  - 6.2.4.5. Hamiltonian Cycle and TSP

Learning Outcomes:

No	Learning Outcomes
LO1	Describe various fundamental data structures, their properties, algorithm design techniques and various means of evaluating algorithms
LO2	Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context.
LO3	Solve problems using Algorithms for Linear and Non-Linear Data Structures, Graph algorithms (Shortest Paths Algorithms, Connectivity and Reachability Algorithms, Spanning Trees)
LO4	Explain with a practical example, each of the algorithm design strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming)
LO5	Use brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming techniques to solve a given algorithm design problem.
LO6	Relate the real-world problems to known data structures and algorithms leading to the recommend appropriate solutions in representation and implementation.
LO7	Explain the significance of NP-completeness
LO8	Classify problems into complexity classes P and NP and to prove hardness of problems

Part B: Contact Session Plan

Academic Term	First Semester 2018-2019
Course Title	Data Structures and Algorithms Design
Course No	SS ZG519
Content Developer	SURENDER SINGH SAMANT

Glossary of Terms

1. Contact Hour (CH) stands for a hour long live session with students conducted either in a physical classroom or enabled through technology. In this model of instruction, instructor led sessions will be for 22 CH.
    a. Pre CH = Self Learning done prior to a given contact hour
    b. During CH = Content to be discussed during the contact hour by the course instructor
    c. Post CH = Self Learning done post the contact hour
2. Contact Hour (CS) stands for a two-hour long live session with students conducted either in a physical classroom or enabled through technology. In this model of instruction, instructor led sessions will be for 11 CS.
    a. Pre CS = Self Learning done prior to a given contact session

   b. During CS = Content to be discussed during the contact session by the course instructor
   c. Post CS = Self Learning done post the contact session
3. RL stands for Recorded Lecture or Recorded Lesson. It is presented to the student through an online portal. A given RL unfolds as a sequences of video segments interleaved with exercises
4. SS stands for Self-Study to be done as a study of relevant sections from textbooks and reference books. It could also include study of external resources.
5. LE stands for Lab Exercises
6. HW stands for Home Work.
7. M stands for module. Module is a standalone quantum of designed content. A typical course is delivered using a string of modules. M2 means module 2.

Teaching Methodology (Flipped Learning Model)

The pedagogy for this course is centered around flipped learning model in which the traditional class-room instruction is replaced with recorded lectures to be watched at home as per the student's convenience and the erstwhile home-working or tutorials become the focus of classroom contact sessions. Students are expected to finish the home works on time.

Contact Session Plan

- Each Module (M#) covers an independent topic and module may encompass more than one Recorded Lecture (RL).
- Contact Sessions **(2hrs each week)** are scheduled alternate weeks after the student watches all Recorded Lectures (RLs) of the specified Modules (listed below) during the previous week
- In the flipped learning model, Contact Sessions are meant for in-classroom discussions on cases, tutorials/exercises or responding to student's questions/clarification--- may encompass more than one Module/RLs/CS topic.
- Contact Session topics listed in course structure (numbered CSx.y) may cover several RLs; and as per the pace of instructor/students' learning, the instructor may take up more than one CS topic during each of the below sessions.

Detailed Structure

Introductory Video/Document: *<< Introducing the faculty, overview of the course, structure and organization of topics, guidance for navigating the content, and expectations from students>>*

- Each of the sub-modules of **Recorded Lectures** (RLx.y) shall delivered via **30 – 60mins videos** followed by:
- **Contact session** (CSx.y) of 2Hr each for illustrating the concepts discussed in the videos with exercises, tutorials and discussion on case-problems (wherever appropriate); contact sessions (CS) may cover more than one recorded-lecture (RL) videos.

Course Contents

Contact Hour 1

Time	Type	Description	Content Reference
Pre CH	RL1.1		
During CH	CH1	CH1.1= Introduction CH1.2 = Algorithms and it's Specification, CH1.3 = Random Access Machine Model, Counting Primitive Operations CH1.4= Notion of best case, average case and worst case	T1: 1.1.1-1.1.3
Post CH	SS1		
	HW1		
	LE1		
	QZ1		
Lab Reference			

Contact Hour 2

Time	Type	Description	Content Reference
Pre CH	RL1.1		
During CH	CH2	CH2.1 =Use of asymptotic notation (Big-Oh Notation, Little-Oh, Omega and Theta Notations) CH2.2 = Correctness of Algorithms	T1: 1.2
Post CH	SS2	T1 – 1.3	
	HW2	T1 - R-1.15, R-1.19	
	LE2		
	QZ2		
Lab Reference			

Contact Hour 3

Time	Type	Description	Content Reference
Pre CH			
During CH	CH3	CH3.1= Analyzing Recursive Algorithms: CH3.2= Recurrence relations, CH3.3= Specifying runtime of recursive algorithms, CH3.4= Solving recurrence equations	T1: 1.1.4, Class Notes

Post CH	SS3		
	HW3		
	LE3		
	QZ3		
Lab Reference			

Contact Hour 4

Time	Type	Description	Content Reference
Pre CH	RL1.2		
During CH	CH4	CH4.1 =Stacks: ADT and Implementation, Applications CH4.2= Queues: Queue ADT and Implementation, Applications	T1: 2.1
Post CH	SS4		
	HW4		
	LE4		
	QZ4		
Lab Reference			

Contact Hour 5

Time	Type	Description	Content Reference
Pre CH	RL1.2		
During CH	CH5	CH5.1 =List: Notion of position in lists CH5.2 = List ADT and Implementation	T1: 2.2
Post CH	SS5		
	HW5	T1 - R-2.1	
	LE5		
	QZ5		
Lab Reference			

Contact Hour 6

Time	Type	Description	Content Reference
Pre CH	RL1.3		
During CH	CH6	CH6.1 =Trees: Terms and Definition, CH6.2= Tree ADT, Applications CH6.3= Binary Trees : Terms and Definition, CH6.4= Properties	T1: 2.3.1, 2.3.2
Post CH	SS6		
	HW6		
	LE6		
	QZ6		
Lab Reference			

Contact Hour 7

Time	Type	Description	Content Reference
Pre CH	RL1.3		
During CH	CH7	CH7.1 =Binary Trees : Representations (Vector Based and Linked), C H7.2= Binary Tree traversal (In Order, Pre Order, Post Order), CH7.3= Applications	T1: 2.3.3
Post CH	SS7		
	HW7		
	LE7		
	QZ7		
Lab Reference			

Contact Hour 8

Time	Type	Description	Content Reference
Pre CH	RL2.2		
During CH	CH8	CH8.1 =Heaps: Definition and Properties, CH8.2 = Representations (Vector Based and Linked), CH8.3 =Insertion and deletion of elements	T1:2.4.1, 2.4.3
Post CH	SS8		
	HW8		

	LE8		
	QZ8		
Lab Reference			

Contact Hour 9

Time	Type	Description	Content Reference
Pre CH	RL2.2		
During CH	CH9	CH9.1 =Heaps: Heap implementation of priority queue, CH9.2 = Heap sort	T1: 2.4.2, 2.4.3
Post CH	SS9	T1 - Section 2.4.2 PQ-Sort	
	HW9		
	LE9		
	QZ9		
Lab Reference			

Contact Hour 10

Time	Type	Description	Content Reference
Pre CH	RL3.1		
During CH	CH10	CH10.1 =Unordered Dictionary :ADT, Applications CH10.2= Hash Tables: Notion of Hashing and Collision (with a simple vector based hash table) CH10.3 = Hash Functions: Properties, Simple hash functions	T1: 2.5.1, 2.5.2, 2.5.3, 2.5.4
Post CH	SS10		
	HW10		
	LE10		
	QZ10		
Lab Reference			

Contact Hour 11

Time	Type	Description	Content Reference
Pre CH	RL3.2		

During CH	CH11	CH11.1 =Methods for Collision Handling: CH11.2 = Notion of Load Factor, CH11.3 = Rehashing, CH11.4 = Open Addressing	T1: 2.5.5
Post CH	SS11		
	HW11		
	LE11		
	QZ11		
Lab Reference			

Contact Hour 12

Time	Type	Description	Content Reference
Pre CH	RL4.1		
During CH	CH12	CH12.1= Ordered Dictionary: ADT, Applications CH12.2 = Binary Search Tree: Motivation with the task of Searching and Binary Search Algorithm, CH12.3 = Properties of BST CH12.4 = Searching an element in BST, Insertion and Removal of Elements	T1: 3.1.1-3.1.5
Post CH	SS12	T1: 3.1.6	
	HW12		
	LE12		
	QZ12		
Lab Reference			

Contact Hour 13

Time	Type	Description	Content Reference
Pre CH			
During CH	CH13	CH13.1 =CH12.1 =Greedy Method: Design Principles and Strategy, CH12.2 =Fractional Knapsack Problem	T1: 5.1.1
Post CH	SS13		
	HW13		
	LE13		

	QZ13		
Lab Reference			

Contact Hour 14

Time	Type	Description	Content Reference
Pre CH			
During CH	CH14	CH14.1 =Greedy Method Task Scheduling Problem	T1:5.1.2
Post CH	SS14		
	HW14		
	LE14		
	QZ14		
Lab Reference			

Contact Hour 15

Time	Type	Description	Content Reference
Pre CH	RL2.1		
During CH	CH15	CH15.1 =Divide and Conquer: Design Principles and Strategy, CH15.2 = Analyzing Divide and Conquer Algorithms CH15.3 = Integer Multiplication Problem	T1:5.2.1, 5.2.2
Post CH	SS15		
	HW15		
	LE15		
	QZ15		
Lab Reference			

Contact Hour 16

Time	Type	Description	Content Reference
Pre CH	RL2.1, RL2.2		
During CH	CH16	CH16.1 = Merge Sort CH16.2 =Quick Sort	T1:4.1, 4.3

Post CH	SS16		
	HW16		
	LE16		
	QZ16		
Lab Reference			

Contact Hour 17

Time	Type	Description	Content Reference
Pre CH			
During CH	CH17	CH17.1 =Dynamic Programming: Design Principles and Strategy, CH17.2 = Matrix Chain Product Problem CH17.3 =0/1 Knapsack Problem	T1: 5.3.1, 5.3.2
Post CH	SS17		
	HW17		
	LE17		
	QZ17		
Lab Reference			

Contact Hour 18

Time	Type	Description	Content Reference
Pre CH	RL5.1, RL5.2		
During CH	CH18	CH18.1 = Graphs : Terms and Definitions, Properties, CH18.2 = Representations (Edge List, Adjacency list, Adjacency Matrix) CH18.3 = Graph Traversals	T1: 6.1, 6.2,6.3
Post CH	SS18	T1: 6.4	
	HW18		
	LE18		
	QZ18		
Lab Reference			

Contact Hour 19

Time	Type	Description	Content Reference
Pre CH	RL 5.4, RL 5.5		
During CH	CH19	CH 19.1 =Single Source Shortest Path algorithm: Dijkstra's Algorithm CH19.2 = Definition of P and NP classes and examples	T1:7.1.1 T1: 13.1
Post CH	SS19	T1.7.1.2, 7.2.1, 7.3.1, 7.3.2	
	HW19		
	LE19		
	QZ10		
Lab Reference			

Contact Hour 20

Time	Type	Description	Content Reference
Pre CH			
During CH	CH20	CH20.1 =Understanding NP-Completeness: CNF-SAT Cook-Levin theorem CH20.2 =Polynomial time reducibility: CNF-SAT and 3-SAT, Vertex Cover	T 13.2, T 13.3
Post CH	SS20		
	HW20		
	LE20		
	QZ20		
Lab Reference			

Contact Hour 21

Time	Type	Description	Content Reference
Pre CH			
During CH	CH21	CH21.1 =Clique and Set-Cover, CH21.2 = Subset-Sum and Knapsack	T1:13.3
Post CH	SS21	Traveling Salesman Problem	
	HW21		

	LE21		
	QZ21		
Lab Reference			

Contact Hour 22

Time	Type	Description	Content Reference
Pre CH			
During CH	CH22	CH22.1 = Course Review	
Post CH	SS22		
	HW22		
	LE22		
	QZ22		
Lab Reference			

Evaluation Scheme:
Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1	Quiz-I/ Assignment-I	Online	-	5%	September 10 to 20, 2018
	Quiz-II			5%	October 20 to 30, 2018
	Quiz-III/ Assignment-II			5%	November 10 to 20, 2018
EC-2	Mid-Semester Test	Closed Book	2 hours	35%	29/09/2018 (AN) 2 PM – 4 PM
EC-3	Comprehensive Exam	Open Book	3 hours	50%	24/11/2018 (AN) 2 PM – 5 PM

Note - *Evaluation components can be tailored depending on the proposed model.*

Important Information:

Syllabus for Mid-Semester Test (Closed Book): Topics in CS 1-5.
Syllabus for Comprehensive Exam (Open Book): All topics given in plan of study
Evaluation Guidelines:

1. For Closed Book tests: No books or reference material of any kind will be permitted. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.

2. For Open Book exams: Use of prescribed and reference text books, in original (not photocopies) is permitted. Class notes/slides as reference material in filed or bound form is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not

allowed.

3. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam. The genuineness of the reason for absence in the Regular Exam shall be assessed prior to giving permission to appear for the Make-up Exam. Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the handout.