School of Information Technology
Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
New Scheme of Examination as per AICTE Flexible Curricula
Syllabus III Semester
B.Tech Computer Science and Engineering (Data Science)

CD-301 (DESIGN THINKING)

	Leadership Oriented Learning (LOL)	

Nature of Course	Behavioral
Pre requisites	Completion of all units from Semesters 1, 2, 3 and 4

Course Terminal Objectives:

1	Recognize the importance of DT
2	Explain the phases in the DT process
3	List the steps required to complete each phase in DT process
4	Apply each phase in the DT process
5	Use doodling and storytelling in presenting ideas and prototypes
6	Create value proposition statements as part of their presentations
7	Recognize how DT can help in functional work
8	Recognize how Agile and DT complement each other to deliver customer satisfaction

Course Enabling Objectives:
Upon completion of the course, students shall have ability to

1	Recognize the importance of Design Thinking	[U]
2	Identify the steps in the DT process	[C]
3	Recognize the steps in the empathize phase of DT	[C]
4	Identify the steps required to conduct an immersion activity	[C]
5	Conduct an immersion activity and fill up the DT question template	[AP]
6	Recognize the steps to create personas in the define phase of DT	[C]
7	Create personas in the define phase of DT	[AP]
8	Recognize the steps to create problem statements in the define phase of DT	[AP]
9	Define the problem statements in the define phase of DT	[E]
10	Recognize the steps in the ideate phase of DT	[C]
11	Apply the steps in the ideate phase of DT	[AP]
12	Recognize how doodling can help to express ideas	[U]
13	Recognize the importance storytelling in presenting ideas and protypes	[U]
14	Recognize the importance of the prototype phase in DT	[C]

15	Create a prototype	[AP]
16	Recognize the importance of service value proposition	[C]
17	Create a value proposition statement	[AP]
18	Recognize the best practices of the testing phase in DT	[U]
19	Test a prototype created through a DT process	[AP]
20	Recognize how DT can help in functional work	[E]
21	Recognize how Agile and DT complement each other to deliver customer satisfaction	[C]

Course Contents:

Total Hours:	45 hours

Textbooks:

	There are no prescribed texts for Semester 5 – there will be handouts and referencelinks shared.

Reference Books:

1	Hooked by NirEyal
2	The Art of Creative Thinking by Rod Judkins
3	Start Up nation by Dan Senor and Saul singer
4	Start with Why by Simon Sinek

Web References:

1	What is Design Thinking? Interaction Design Foundation
2	What are some of the good examples of design thinking? - Quora
3	Design thinking 101: Principles, Tools & Examples to transform your creative process

Online Resources:

1	Understanding Design thinking WF NEN
2	Design Thinking and Innovation at Apple Wei Li
3	Stanford Webinar- Design Thinking = Method, Not Magic
4	Stanford Design Thinking Virtual Crash Course
5	So Many Uses- activity to spark creativity and design

Assessment Methods & Levels (based on Bloom's Taxonomy)

Formative assessment (Max. Marks:20)

Course Outcome	Bloom's Level	Assessment Component	Marks
	Apply	Defining problem statement	5
	Apply	Ideating solutions	5
	Apply	Creating a prototype	10

Summative Assessment based on End Semester Project

Bloom's Level		
Understand	Understand, Analyze, Apply	50

Apply Analyze	Conduct and apply DT in the project.	

Lesson Plan

Unit No	Objective	Bloom's Level	Content	Type of Class	Duration
1	Recognize the importance of Design Thinking	2	**Why is Design Thinkingimportant for business?** Stories and examples will be usedto introduce Design Thinking to the participants. We will use relevant stories and the followingvideos. 1. YouTube video: The DesignThinking Process – Sprouts(3.57 mins) 2. Leverage TCS-provided DT content to show the evolution of DT and why isimportant in present business environment. Canbe a video. (2 mins) Lecturer to encourage the students to maintain their Satorislam book and capture their learning points in it.	Introduction and discussion	60 mins
1	Recognize the importance of Design Thinking	2	**Why is Design Thinking importantfor you?** Experiential activity **Products that you loved and**	Activity	90 mins
Unit No	**Objective**	**Bloom's Level**	**Content**	**Type of Class**	**Duration**
			hated: In this activity, learners willhave to share about a product theylike of disliked based on their experience. What would they need in a badproduct to make it good?		

1	Identify the steps in the DT process	2	**What is DT?** Introduce the 5-Step Stanford Model using YouTube videos: The video will give a brief idea about the five steps: • Empathize (search for rich stories and find some love) • Define (user need and insights – their POV) • Ideate (ideas, ideas, ideas) • Prototype (build to learn) • Test (show, don't tell) Start all over and iterate the flow as much as possible	Lecture and demo	60 mins
1	Recognize the steps in the empathize phase of DT	2	**What is empathy?** Touch the target activity (Recap from Sem 2 Unit 4) Discussions in class Reference: FHIL | Stages of Design Thinking | EMPATHY (2:29 mins)	Activity	60 mins
1	Identify the steps required to conduct an immersion activity	1 and 2	**How to empathize?** Moccasin Walk activity for 1 hour to allow learners experience stepping into the shoes of another person. *This is an individual activity.* Sharing observations with the group. Suggest that students try this even in their free time away from studies.	Activity and lecture	90 mins
1	Identify the steps required to conduct an immersion activity	1 and 2	**Intro to Immersion Activity** Introduction to immersion activity through flowcharts and handouts and examples (to be provided by	Lecture	45 mins
Unit No	**Objective**	**Bloom's Level**	**Content**	**Type of Class**	**Duration**
			TCS DT Team) (steps and the question template: 1. We met; 2. We were amazed to realize that; 3. We wonder if this means 4. It would change the world if)		

1	Conduct an immersion activityand fill up the DT question template	3	**Immersion activity** Participants will be divided into four groups. Each group will needto visit any one of the following places to conduct an immersion activity. They need to interview people and fill up the DT questiontemplate (explained in the last class) 1. College cafeteria 2. College library 3. College sports facility 4. Transport facility nearcollege	Practical	180 mins
2	Recognize the steps to create personas in thedefine phase ofDT Create personas inthe define phase of DT	2 3	**Creating personas** Start with YouTube videos explaining the process of personacreation: 1. Personas – What is a persona and how do Icreate one? (2019) https://www.youtube.com/watch?v=GNvLpfXCge8 Each group will create at least onepersona based on the immersion study they conducted in the empathize stage (refer to the fourquestion templates). The group can use A4 pages, colours and other props to create and display their respective persona. **Reference:** https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use- them	Lecture and practical	120 mins
Unit No	**Objective**	**Bloom's Level**	**Conten t**	**Type of Class**	**Duratio n**
			Lecturer to guide participants on getting the personas right (based on guidelines provided by TCS DTTeam).		
2	Recognize the steps to create problem statements in thedefine phase of DT	2	**Problem statements** Session will begin with YouTube videos on how to define problemstatements in the Define phase. 1. FHIL | Stages of Design Thinking | REFRAME (1:55mins) Lecturer will provide examples	Lecture and demo	60 mins

			of problem statements in class (basedon handouts provided by TCS DT Team)		
2	Define the problem statements in thedefine phase of DT	3	**Defining problem statements** Group activity, in which each groupwill define the key problem statements (max three) for their lead personas. Each group will present while theremaining groups will do a peer review. Finally, lecturer will moderate/validate the problemstatements (based on handoutsprovided by TCS DT Team)	Formative assessmen t	90 mins
3	Recognize the steps in the ideatephase of DT	1 and 2	**How to Ideate?** The session will start with YouTubevideos:	Lecture and demo	60 mins
Unit No	**Objective**	**Bloom's Level**	**Conten t**	**Type of Class**	**Duratio n**
			1. FHIL | Stages of DesignThinking | IDEATE (1:54secs) 2. What Is Six Thinking Hats?(Litmos Heroes) (1:58 secs) Lecturer to briefly tell them about the guidelines of ideating (to beprovided by TCS DT Team)		
3	Apply the steps inthe ideate phase of DT	3	**Ideation games** Game 1: Six Thinking Hats Game 2: Million- dollar idea	Activity	90 mins
3	Apply the steps inthe ideate phase of DT	3	**Ideate to find solutions** Participants will work in their assigned groups to ideate solutionsfor the problem statements they identified (as continuation of immersion activity) applying ideation methods discussed in the previous session. They will get scores based on how well they canapply the ideation methods. Lecturers will observe the groups separately and assign them scoresbased on specific rubric (provided by the TCS DT Team).	Formative assessmen t	90 mins

3	Recognize how doodling can helpto express ideas	1	**Let's doodle!** Participants will first watch a videoon doodling: Doodling – how it can help in presenting ideas during ideate andprotype phases After that, participants will complete an activity on doodling.	Demo and activity	60 mins
3	Recognize the importance storytelling in presenting ideasand protypes	1	**What is Storytelling in DT?** Activity- Research to find out aboutpeople who have used DT in providing solutions. Present their findings in forms of stories. (Recapfrom Unit- Sem-) Suggested topics to be provided bythe TCS DT team.	Activity	120 mins
Unit No	**Objective**	**Bloom's Level**	**Content**	**Type of Class**	**Duration**
4	Recognize the importance of theprototype phase in DT	2	**Why is a Prototype important inDesign Thinking?** The session will start with an activity to drive home the importance of creating a prototypein the design thinking process. As part of debrief of the activity, lecturer will share relevant examples and prototyping guidelines (provided by the TCS DT Team). Finally, the participants will watchtwo YouTube videos: 1. FHIL | Stages of Design Thinking | PROTOTYPE 2. Prototyping Phase - Design Thinking | Coursera https://www.coursera.org/lecture/patient-safety-project-planning/prototyping-phase-jVuQn	Activity and demo	60 mins

4	Create a prototype	3	**Prototype your idea** This is a group activity in which theparticipants will work in groups (created at the beginning of the course, in which they did immersion, persona creation, defining problem statement and ideating) to create prototypes based on the solutions they had identified. Lecturer to share feedback based on guidelines provided by the TCsDT team.	Formative assessment	180 mins
4	Recognize the importance ofservice value proposition Create a value proposition statement	2 3	**Value Proposition Statement** You Tube: What is Value Proposition (by Venture Well)(3:51 mins)? Lecturer to discuss the guidelinesfor creating a value proposition	Lecture	120 mins 1635 mins
Unit No	**Objective**	**Bloom's Level**	**Content**	**Type of Class**	**Duration**
			statement (to be provided by theTCS DT Team) Each group now needs to create value proposition statement for the solution they have suggested.		
4	Recognize the best practices ofthe testing phasein DT	1	**Testing in Design Thinking** Participants will first watch aYouTube video: FHIL | Stages of Design Thinking |TESTING After that lecturers will explainthem the importance of Testingthe prototype through stories (provided by the TCS DT Team). They will also explain how the loop works in DT between the Empathize and Testing phases.	Lecture	60 mins

	Test a prototype created through aDT process	3	**Test the Prototype** Each group needs to test theirprototype created earlier and: 1. Document user feedback 2. Write down their inferencefrom the feedback 3. Suggest next steps (the loop that happens in DT)	Activity	120 mins
4	Recognize how DTcan help in functional work	1	**Role of DT in your work** Lecturer conducts a group/openhouse discussion on: "How DT can help me to become abetter coder?" Lecturer needs to capture the key learning points in thesediscussions.	Discussion	60 mins
4	Recognize how Agile and DT complement eachother to deliver customer satisfaction	1	Suggested session on: **How Agile and DT complement each other to deliver customer satisfaction**	Lecture	45 mins
4			**Share your Satori** Participants will be asked to share their Satori moments from the DTsessions	Reflection activity	60 mins
Unit No	**Objective**	**Bloom's Level**	**Conten t**	**Type of Class**	**Duratio n**
					33 hours
			Project **Option 1:** Each group needs to present a Prototype of how theycan apply DT in their functional work or coding. Examples will beprovided to explain what exactlythey need to do. Option 2: Each group will apply DTto create a prototype to improve any existing product or service. For both options, groups need to complete all phases of the Stanford DT model and include theoutputs of each phase in their presentation. Lecturers will evaluate the projectbased on the rubric provided by the TCS DT Team.		12 hours
				Total	**45 hours**

CD-302 (COMPUTER ORGANIZATION AND ARCHITECTURE)

UNIT I- Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.
Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.
Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT II- Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.
Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT III- Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.
Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT IV- Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.
Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT V- Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.
Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.
Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

Reference Books-

1. Computer System Architecture – M. Moris Mano, Third Edition, Pearson/PHI.
2. Computer Organization – Car Hamacher, ZvonksVranesic, SafeaZaky, Vth Edition, McGraw Hill.
3. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
4. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson

Course Outcomes-

On successful completion of the course, the students will be able to:
CO1: Understand the basics of instructions sets and their impact on processor design.
CO2: Demonstrate an understanding of the design of the functional units of a digital computer system.
CO3: Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
CO4: Design a pipeline for consistent execution of instructions with minimum hazards.
CO5: Recognize and manipulate representations of numbers stored in digital computers

CD-303 (DATA STRUCTURES AND ALGORITHMS)

Unit I Introduction: Data, data type, data object. Types of data structure – primitive & nonprimitive, linear & non-linear. Operations on data structures – traversing, searching, inserting, deleting, Complexity analysis – worst case, best case, and average case. Time – space trade off, algorithm efficiency, asymptotic notations – big oh , omega , theta.

Unit II Arrays & Structure: Introduction , declaration of arrays , operations on arrays – inserting , deleting , merging of two arrays , 1 dimensional & 2 dimensional arrays, row & column major representation , address calculation in array , storing values in arrays , evaluation of polynomial – addition & representation. Searching & sorting – Introduction, sequential search, binary search, Fibonacci search, indexed sequential search, hashed search. Types of sorting with general concepts – bubble, heap, insertion, selection, quick, heap, shell, bucket, radix and merge
sort.

Unit III Stacks & Queues Basic concept of stacks & queues, array representation of stacks, operation on stacks – push , pop , create , getTop , empty , linked representation of stack ,multiple stack. Application of stack – Conversion: infix, prefix, postfix and evaluation of arithmetic expression. Linked representation of queue, operations on queue – insertion &deletion. Types of queue with functions – circular, deque, and priority queue. Applications of queues – job scheduling, Josephus problem.

Unit IV Linked List Introduction – basic terminology, memory allocation & deallocation for linked list. Linked list variants – head pointer, head node, types linked list – linear & circular linked list. Doubly linked list, creation of doubly list, deletion of node from doubly linked list, insertion of a node from doubly linked list, traversal of doubly linked list. Circular linked list – singly circular linked list , circular linked list with header node , doubly circular linked list. Applications of linked list – polynomial representation & garbage collection.

Unit V Trees Basic terminology – general tree , representation of general tree, types of trees, binary tree- realization and properties , traversal in binary trees – inorder , preorder , postorder , applications of trees. Graph- Basic Terminologies and representations, Graph search and traversal algorithms.

Reference Books-

1.Varsha H. Patil "Data Structure Using C++" Oxford.
2. ReemaThareja" Data Structure Using C " Oxford.
3. D. S Malik "Data Structure Using C++ " Second Edition Cengage.
4. Kushwaha and Mishra "Data Structure: A programming Approach with C", PHI Learning.
5. A. K Sharma "Data Structure Using C" Pearson.
6. Ellis Horowitz, SartajSahni, "Fundamentals of Data Structures", Computer Science Press

Course Outcomes-

After the completion of this course, the students will be able to:
CO1 Implement searching algorithms (linear search and binary search) on given data
CO2 Perform operations on arrays
CO3 Implement programs using queues, stacks and link lists
CO4 Implement sorting operation using various algorithms and compare their performance
CO5 Implement tree, graph search and traversal algorithms

Suggested List of Experiments-

1. Write a program to search an element in the array using Linear and Binary Search.
2. Write a program to perform the following operation in Matrix:
a. Addition b. Subtraction c. Multiplication d. Transpose
3. Write a program to perform the following operation on strings using string functions:
a. Addition b. Copying c. Reverse d. Length of String
4. Write program for implementing the following sorting methods to arrange a list of integers in

ascending order:
a) Quick sort b) Selection sort c) Insertion sort d) Merge sort
5. Write a program that uses stack operations to convert a given infix expression into its postfix equivalent.
6. Write a program to merge two sorted array into one sorted array.
7. Write a program to implement stack using array and linked list.
8. Write a program to implement queue and circular queue using array.
9. Write a program to insert an element in the beginning and end of singly linked list.
10. Write a program to insert an element at any position in singly and doubly linked list.
11. Insert and delete a node at any position in doubly linked list.
12. Write a program of Tower of Hanoi.
13. Write a program that uses functions to perform the following:
a) Create a binary search tree of integers.
b) Traverse the above Binary search tree non recursively in in order.

CD-304 (DATABASE MANAGEMENT SYSTEMS)

Unit I Basic Concepts: Introduction to DBMS, File system vs DBMS, Advantages of database systems, Database System architecture, Data models, Schemas and instances, Data independence, Functions of DBA and designer, Entities and attributes, Entity types, Key attributes, Relationships, Defining the E-R diagram of database.

Unit II Relational Model: Structure of relational databases, Domains, Relations, Relational algebra – fundamental operators and syntax, relational algebra queries, Entity-Relationship Model: Basic concepts, Design process, constraints, Keys, Design issues, E-R diagrams, weak entity sets, extended E-R features – generalization, specialization and aggregation

Unit III SQL: Data definition in SQL, update statements and views in SQL: Data storage and definitions, Data retrieval queries and update statements, Query Processing & Query Optimization: Overview, measures of query cost, selection operation, sorting, join, evaluation of expressions, transformation of relational expressions, estimating statistics of expression results, evaluation plans. Case Study of ORACLE and DB2.

Unit IV Relational Database design: Functional Dependency –definition, trivial and non-trivial FD, closure of FD set, closure of attributes, irreducible set of FD, Normalization –1NF, 2NF, 3NF, Decomposition using FD-dependency preservation, lossless join, BCNF, Multi-valued dependency, 4NF, Join dependency and 5NF

Unit V Introduction of transaction, transaction processing and recovery, Concurrency control: Lock management, specialized locking techniques, concurrency control without locking, Protection and Security Introduction to: Distributed databases, Basic concepts of object oriented data base system.

Reference Books-

1. Korth, Silbertz, Sudarshan, "Database Concepts", McGrawHill.
2. Elmasri, Navathe, "Fundamentals of Database Systems", Pearson.
3. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB publications.
4. S. Sharma, J. Agrawal, S. Agrawal, "Advanced Database Management System", Dreamtech Press.
5. Leon & Leon, "Fundamental of Data Base Management System",TMH

Course Outcomes-

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

CO1 Analyze the physical and logical database designs, database modeling, relational, hierarchical, and network models

CO2 Use DDL, DML and DCL utilities to implement the schema using a DBMS.

CO3 Formulate data retrieval queries in SQL and Relational Algebra.

CO4 Demonstrate an understanding of functional dependencies, normalization theory and apply such knowledge to the design of database.

CO5 Understand concepts of Transaction Processing, Concurrency Control, distributed database and big data.

Suggested List of Experiments-

1. To perform various SQL Commands of DDL, DML, DCL.
2. Write SQL Commands such as Insertion, deletion and updation for any schema.
3. To execute Nested Queries, Join Queries, order-by, having clause and string operation.
4. To perform set operators like Union, Intersect, Minus on a set of tables.
5. To execute various commands for GROUP functions (avg, count, max, min, Sum).
6. Write a PL/SQL block for transaction application usingTriggers.
7. Write a DBMS program to prepare report for an application using function.
8. Designing of various Input screens/Forms.
9. Create reports using database connectivity of Front end with back end.
10. Create database Design with normalization and implementing in any application.

CD-305 (INTRODUCTION TO DATA SCIENCE)

Unit I

What is Data Science; Emergence and Evolution of Data Science; Where do we see Data Science; How does Data Science relate to other fields; Information vs Data, Skills for Data Science, Tools for Data Science, Issues of Ethics, Bias and Privacy in Data Science; Data Scientist roles and responsibilities, Applications of Data Science in various fields, Business Intelligence versus Data Science,

Unit II

Type of Data: Structured, Semi Structured, UnStructured and their examples, Categorical/Nominal/Ordinal data, Data Types Conversion, Challenges with unstructured data,
Data Collection Methods, - Open data, social media data, Multimodal data, Data storage and presentation
Data Preprocessing- Data Cleaning, Data Integration, Data Transformation, Data Reduction, Data Discretization

Unit III

Data Analysis and Data Analytics, Descriptive Analysis, Diagnostic Analytics, Predictive Analytics, Prescriptive Analytics, Exploratory Analysis, Mechanistic Analysis, Regression, Data Visualization

Unit IV

Data Science Project Lifecycle, Standard Data Science Tasks, Tools for Data Science, Introduction of Python, R, MySQL

Unit V

Introduction to Machine Learning for Data Science, Supervised, Unsupervised and Reinforcement learning, Examples and techniques

Reference Books-

1. JojoMoolayil, “Smarter Decisions : The Intersection of IoT and Data Science”,PACKT, 2016.
2. Cathy O’Neil and Rachel Schutt , “Doing Data Science”, O'Reilly, 2015.
3. David Dietrich, Barry Heller, Beibei Yang, “Data Science and Big data Analytics”,EMC 2013
4. Raj, Pethuru, “Handbook of Research on Cloud Infrastructures for Big DataAnalytics”, IGI Global.
5. Chirag Shah, “A Hands-on Introduction to Data Science”, Cambridge University Press
6. John D Kelleher, Brendan Tierney, “Data Science”, MIT Press

Course Outcomes-

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

CO1. Explain the applications of Data Science
CO2. Understand various categories of data and the operations on data
CO3. Analyse the data to find patterns
CO4. Understand the life cycle of a Data Science project
CO5**.** Apply machine learning techniques for learning from data

CD 306 (COMPUTER PROGRAMMING III)
PYTHON

Unit I

Introduction to python language, Basic syntax, Literal Constants, Numbers, Variable and Basic data types,String, Escape Sequences, Operators and Expressions, Evaluation Order, Indentation, Input, Output, Functions, Comments.

Unit II

Data Structure: List, Tuples, Dictionary, DataFrame and Sets, constructing, indexing, slicing and content manipulation.

Unit III

Control Flow:Conditional Statements - If, If-else, Nested If-else. Iterative Statement - For, While, Nested Loops. Control statements - Break, Continue, Pass.

Unit IV

Object oriented programming:Class and Object, Attributes, Methods, Scopes and Namespaces, Inheritance, Overloading, Overriding, Data hiding,Exception: Exception Handling, Except clause, Try finally clause, User Defined Exceptions.

Unit V

Modules and Packages: Standard Libraries: File I/0, Sys, logging, Regular expression, Date and Time, Network programming, multi-processing and multithreading.

Reference Books-

1. Timothy A. Budd: Exploring python, McGraw-Hill Education.
2. R.NageshwarRao ,"Python Programming" ,Wiley India
3. Think Python: Allen B. Downey, O'Reilly Media, Inc.

Suggested List of Experiments

1. To write a Python program to find GCD of two numbers.
2. To write a Python Program to find the square root of a number by Newton's Method.
3. To write a Python program to find the exponentiation of a number.
4. To write a Python Program to find the maximum from a list of numbers.
5. To write a Python Program to perform Linear Search
6. To write a Python Program to perform binary search.
7. To write a Python Program to perform selection sort.
8. To write a Python Program to perform insertion sort.
9. To write a Python Program to perform Merge sort.
10. To write a Python program to find first n prime numbers.
11. To write a Python program to multiply matrices.
12. To write a Python program for command line arguments.
13. To write a Python program to find the most frequent words in a text read from a file.
14. To write a Python program to simulate elliptical orbits in Pygame.
15. To write a Python program to bouncing ball in Pygame.

School of Information Technology
Rajiv Gandhi ProudyogikiVishwavidyalaya, Bhopal
New Scheme of Examination as per AICTE Flexible Curricula
IV Semester Bachelor of Technology (B.Tech.)
Computer Science and Engineering (Data Science)
Syllabus

CD 401 (THEORY OF COMPUTATION)

UNIT I: Fundamentals: Strings, Alphabet, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, deterministic finite automaton and nondeterministic finite automaton, transition diagrams and Language recognizers.
Finite Automata: NFA with Î transitions - Significance, acceptance of languages. Conversions & Equivalence: Equivalence between NFA with and without Î transitions, NFA to DFA conversion, minimization of FSM, equivalence between two FSM's, Finite Automata with output- Moore and Melay machines.

UNIT II: Regular Languages: Regular sets, regular expressions, identity rules, Constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets.
Grammar Formalism: Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar, derivation trees, and sentential forms. Right most and left most derivation of strings.

UNIT III: Context Free Grammars: Ambiguity in context free grammars. Minimization of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of CFL.
Push Down Automata: Push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion. (Proofs not required). Introduction to DCFL and DPDA.

UNIT IV: Turing Machine: Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines (proofs not required). , linear bounded automata and context sensitive language

UNIT V: Computability Theory: Chomsky hierarchy of languages, decidability of, problems, Universal Turing Machine, undecidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

Reference Books-

1. "Introduction to Automata Theory Languages and Computation". Hopcroft H.E. and Ullman J. D. Pearson Education.
2. Introduction to Theory of Computation –Sipser 2nd edition Thomson.
3. Introduction to Formal Languages, Automata Theory and Computation Kamala Krithivasan, Rama
4. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
5. Theory of Computation : A Problem – Solving Approach- Kavi Mahesh, Wiley India Pvt. Ltd.
6. "Elements of Theory of Computation", Lewis H.P. &Papadimition C.H. Pearson /PHI.
7. Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 2nd edition, PHI.
8. Introduction to languages and the Theory of Computation, John C Martin, TMH.

Course Outcomes-

On successful completion of the course, the students will be able to:

CO1: Explain basic concepts in formal language theory, grammars, automata theory, computability theory, and complexity theory.

CO2: Demonstrate abstract models of computing, including deterministic (DFA), non-deterministic (NFA), Push Down Automata(PDA) and Turing (TM) machine models and their power to recognize the languages.

CO3: Explain the application of machine models and descriptors to compiler theory and parsing. Students will be able to relate practical problems to languages, automata, computability, and complexity.

CO4: Demonstrate an increased level of mathematical sophistication.

CO5: Apply mathematical and formal techniques for solving problems in computer science. Students will be able to explain the relationship among language classes and grammars with the help of Chomsky Hierarchy

CD 402 (OPERATING SYSTEMS)

UNIT I- Introduction: Concept of Operating Systems (OS), Generations of OS, Types of OS, OS Services, Interrupt handling and System Calls, Basic architectural concepts of an OS, Concept of Virtual Machine, Resource Manager view, process view and hierarchical view of an OS.

UNIT II- Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time. Scheduling algorithms: Pre-emptive and non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling; Real Time scheduling: RM and EDF.

UNIT III - I/O Hardware: I/O devices, Device controllers, Direct Memory Access, Principles of I/O.
File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed),Free-space management (bit vector, linked list, grouping), directory implementation(linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

UNIT IV- Inter-process Communication: Concurrent processes, precedence graphs, Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Semaphores, Strict Alternation, Peterson's Solution, The Producer / Consumer Problem, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem, Barber's shop problem.
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery. Concurrent Programming: Critical region, conditional critical region, monitors, concurrent languages, communicating sequential process (CSP)

UNIT V- Memory Management: Basic concept, Logical and Physical address maps, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page allocation, Partitioning, Paging, Page fault, Working Set, Segmentation, Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).
Case study: UNIX/Linux OS file system, shell, filters, shell programming, programming with the standard I/O, UNIX/Linux system calls.

Reference Books-

1. Operating System Concepts Essentials. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne.
2. Operating Systems: Internals and Design Principles. William Stallings.
3. Operating System: A Design-oriented Approach. Charles Patrick Crowley.
4. Operating Systems: A Modern Perspective. Gary J. Nutt.
5. Design of the Unix Operating Systems. Maurice J. Bach.
6. Understanding the Linux Kernel, Daniel Pierre Bovet, .itaseC ocraM

Course Outcomes-

On successful completion of the course, the students will be able to:

CO1: Understand the various OS functionalities and acquire the knowledge of various types of OS

CO2: Design and Implement CPU scheduling algorithms to meet and validate the scheduling criteria

CO3: Implement directories and perform various operations on files/directories

CO4: Apply the acquired knowledge of deadlocks to design and implement deadlock free computer programs as well as understand the issues in inter process communication

CO5: Understand how memory is allocated to processes by OS and Implement algorithms related to main and Virtual memory techniques

Suggested List of Experiments-

1. Unix/Linux commands (files directory, data manipulation, network communication etc), shell programming and vi editor
2. C programs for implementation of the following:
    a. Scheduling Algorithms
    b. Shared memory
    c. Thread and Multi Thread
    d. Inter Process Communication
    e. Deadlock Avoidance and Deadlock Detection
    f. Semaphore
    g. Memory Management
    h. Indexing and Hashing
3. C Programs for implementing certain commands and a shell like Unix/Linux system shell, using the Unix/Linux System calls.

CD 403 (ANALYSIS & DESIGN OF ALGORITHMS)

Unit I Algorithms, Designing algorithms, analyzing algorithms, asymptotic notations, heap and heap sort. Introduction to divide and conquer technique, analysis, design and comparison of various algorithms based on this technique, example binary search, merge sort, quick sort, strassen's matrix multiplication

Unit II Study of Greedy strategy, examples of greedy method like optimal merge patterns, Huffman coding, minimum spanning trees, knapsack problem, job sequencing with deadlines, single source shortest path algorithm, etc.

Unit III Concept of dynamic programming, problems based on this approach such as 0/1 knapsack, multistage graph, reliability design, Floyd-Warshall algorithm, etc.

Unit IV Backtracking concept and its examples like 8 queen's problem, Hamiltonian cycle, Graph coloring problem etc. Introduction to branch & bound method, examples of branch and bound method like traveling salesman problem etc. Meaning of lower bound theory and its use in solving algebraic problem, introduction to parallel algorithms

Unit V Binary search trees, height balanced trees, 2-3 trees, B-trees, basic search and traversal techniques for trees and graphs (Inorder, preorder, postorder, DFS, BFS), NP-completeness

Reference Books-

1. Coremen Thomas, Leiserson CE, Rivest RL; Introduction to Algorithms; PHI.
2. Horowitz & Sahani; Analysis & Design of Algorithm
3. Dasgupta; algorithms; TMH
4. Ullmann; Analysis & Design of Algorithm;
5. Michael T Goodrich, Robarto Tamassia, Algorithm Design, Wiely India

Course Outcomes-

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

1 Implement sorting and searching algorithm
2 Experiment with techniques for obtaining maximum output with minnium efforts
3 Make use of dynamic programming for finding
4 Solve 8 queen's problem and others of the kind for application in real world scenarios .
5 Distinguish between NP hard and NP complete problems and develop their solutions

Suggested List of Experiments-

1. Write a program for Iterative and Recursive Binary Search.
2. Write a program for Merge Sort.
3. Write a program for Quick Sort.
4. Write a program for Strassen's Matrix Multiplication.
5. Write a program for optimal merge patterns.
6. Write a program for Huffman coding.
7. Write a program for minimum spanning trees using Kruskal's algorithm.
8. Write a program for minimum spanning trees using Prim's algorithm.
9. Write a program for single sources shortest path algorithm.
10. Write a program for Floye-Warshal algorithm.
11. Write a program for traveling salesman problem.
12. Write a program for Hamiltonian cycle problem.

CD 404 (SOFTWARE ENGINEERING)

UNIT I- Introduction: Programming in the small vs. programming in the large; software project failures and importance of software quality and timely availability; of software engineering towards successful execution of large software projects; emergence of software engineering as a discipline, Software Engineering Historical Development from Jackson Structured Programming to Agile Development.

UNIT II- Software Project Management: Basic concepts of life cycle models – different models and milestones; software project planning –identification of activities and resources; concepts of feasibility study; techniques for estimation of schedule and effort; software cost estimation models and concepts of software engineering economics; techniques of software project control and reporting; introduction to measurement of software size; introduction to the concepts of risk and its mitigation; configuration management.

UNIT III- Software Quality Management and Reliability: Software quality; Garvin's quality dimensions, McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction to Capability Maturity Models (CMM and CMMI); Introduction to software reliability, reliability models and estimation.

UNIT IV- Software Requirements Analysis, Design and Construction: Introduction to Software Requirements Specifications (SRS) and requirement elicitation techniques; techniques for requirement modelling – decision tables, event tables, state transition tables, Petri nets; requirements documentation through use cases; introduction to UML, introduction to software metrics and metrics-based control methods; measures of code and design quality.
Object Oriented Analysis, Design and Construction: Concepts -- the principles of abstraction, modularity, specification, encapsulation and information hiding; concepts of abstract data type; Class Responsibility Collaborator (CRC) model; quality of design; design measurements; concepts of design patterns; Refactoring; object-oriented construction principles; object oriented metrics.

UNIT V- Software Testing: Introduction to faults and failures; basic testing concepts; concepts of verification and validation; black box and white box tests; white box test coverage – code coverage, condition coverage, branch coverage; basic concepts of black-box tests – equivalence classes, boundary value tests, usage of state tables; testing use cases; transaction based testing; testing for non-functional requirements – volume, performance and efficiency; concepts of inspection; Unit Testing, Integration Testing, System Testing and Acceptance Testing.

Agile Software Engineering: Concepts of Agile Methods, Extreme Programming; Agile Process Model - Scrum, Feature; Scenarios and Stories

Reference Books-

1. Software Engineering, Ian Sommerville
2. Software Engineering A Practitioner's Approach, Roggers S. Pressman and Bruce R. Maxim.
1. The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!, Ivar Jacobson, Harold "Bud" Lawson, Pan-Wei Ng, Paul E. McMahon and Michael Goedicke.
2. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi and Mandrioli Dino.
3. Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices, Michael Jackson.
4. The Unified Development Process, Ivar Jacobson, Grady Booch and James Rumbaugh.
5. Design Patterns: Elements of Object-Oriented Reusable Software, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides.
6. Software Metrics: A Rigorous and Practical Approach, Norman E Fenton and Shari Lawrence Pfleeger.
7. Software Engineering: Theory and Practice, Shari Lawrence Pfleeger and Joanne M. Atlee.
8. Object-Oriented Software Construction, Bertrand Meyer.
9. Object Oriented Software Engineering: A Use Case Driven Approach --Ivar Jacobson.
10. Touch of Class: Learning to Program Well with Objects and Contracts --Bertrand Meyer.

11. UML Distilled: A Brief Guide to the Standard Object Modeling Language --Martin Fowler.
12. Introduction to Business Domains for Software Engineers, Manoj Kumar Lal
13. Knowledge Driven Development – Bridging Waterfall and Agile Methodologies –- Manoj Kumar Lal

Course Outcomes-

On successful completion of the course, the students will be able to:

CO1: Understand software engineering process and practices
CO2: Distinguish between various software process models and cost estimation models and choose appropriate model for project
CO3: Understand various software quality attributes and software reliability metrics
CO4: Develop the SRS document for project
CO5: Apply the knowledge of testing techniques and strategies to test developed software

Suggested List of Experiments-

Development of requirements specification, function-oriented design using SA/SD, object-oriented design using UML, test case design, implementation using C++ and testing. Use of appropriate CASE tools and other tools such as configuration management tools, program analysis tools in the software life cycle.

CD 405 (DATA MINING AND WAREHOUSING)

COURSE OBJECTIVES: Student should understand the value of Historical data and data mining in solving real-world problems. Student should become affluent with the basic Supervised and unsupervised learning algorithms commonly used in data mining . Student develops the skill in using data mining for solving real-world problems.

Unit I

Data Warehousing: Introduction, Delivery Process, Data warehouse Architecture, Data Preprocessing: Data cleaning, Data Integration and transformation, Data reduction. Data warehouse Design: Dataware house schema, Partitioning strategy Data warehouse Implementation, Data Marts, Meta Data, Example of a Multidimensional Data model. Introduction to Pattern Warehousing.

Unit II

OLAP Systems: Basic concepts, OLAP queries, Types of OLAP servers, OLAP operations etc. Data Warehouse Hardware and Operational Design: Security, Backup And Recovery,

Unit III

Introduction to Data& Data Mining :Data Types, Quality of data, Data Preprocessing, Similarity measures, Summary statistics, Data distributions, Basic data mining tasks, Data Mining V/s knowledge discovery in databases. Issues in Data mining. Introduction to Fuzzy sets and fuzzy logic.

Unit IV

Supervised Learning: Classification: Statistical-based algorithms, Distance-based algorithms, Decision tree-based algorithms, Neural network-based algorithms, Rule-based algorithms, Probabilistic Classifiers

Unit V

Clustering & Association Rule mining : Hierarchical algorithms, Partitional algorithms, Clustering large databases – BIRCH, DBSCAN, CURE algorithms.Association rules : Parallel and distributed algorithms such as Apriori and FP growth algorithms.

Text Books:

1. Pang – ningTan , Steinbach & Kumar, “Introduction to Data Mining”, Pearson Edu, 2019.
2. Jaiwei Han, Micheline Kamber, “Data Mining : Concepts and Techniques”, Morgan Kaufmann Publishers.

Reference Books:

1. Margaret H. Dunham, “Data Mining : Introductory and Advanced topics”, Pearson Edu., 2009.
2. Anahory& Murray, “Data Warehousing in the Real World”, Pearson Edu., 2009.

Course Outcomes:

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

CO1. Understand the need of designing Enterprise data warehouses andwill be enabled to approach business problems analytically by identifying opportunities to derive business.

CO2. Compare and contrast, various methods for storing & retrieving data from different data sources/repository.

CO3. Ascertain the application of data mining in various areas and Preprocess the given data and visualize it for a given application or data exploration/mining task

CO4. Apply supervised learning methods to given data sets such as classification and its various types.

CO5. Apply Unsupervised learning methods to given data sets such as clustering and its various types.

CO6 Apply Association rule Mining to various domains.

CD-406 (COMPUTER PROGRAMMING IV)
(JAVA)

Unit I-Overview of Java, Installation, First Simple Program, Compilation process, JavaKeywords,Identifiers, Literals, Comments, Data Types, Variables, Dynamic initialization, typeconversion and casting, Operators, Control Statements.

Unit II-Declaring Objects, Introducing Methods, Constructors, this Keyword, GarbageCollection, finalize Method, Overloading Methods, Overloading Constructors, Using Objects asParameters, Inheritance, Creating a Multilevel Hierarchy, Packages and Interfaces, ExceptionHandling, Multithreaded

Unit III-The Applet Class: Applet Basics, The Applet Class, Applet Architecture, AppletInitialization and Termination , Simple Applet Display Methods, Simple Banner Applet, Usingthe Status Window, The HTML APPLET Tag, Passing Parameters to Applets, Improving theBanner Applet.

Unit IV-Introducing the AWT: Working with Windows, Graphics, and Text, AWT Classes,Window Fundamentals, Component, Container, Panel, Frame, Working with Frame Windows,Handling Events in a Frame Window, AWT Controls, Layout Managers, and Menus, Addingand Removing Controls, Grid Layout, Border Layout, introduction to swing and servlet.

Unit V-Event Handling, Two Event Handling Mechanisms, The Delegation Event Model,Events, Event Sources, Event Listeners, Event Classes, The Mouse Event Class and others,JDBC: JDBCODBC bridge, the connectivity model, the driver manager, navigating the result setobject contents, the JDBC exceptional classes, connecting to remote database.

References:

1. E. Balagurusamy, "Programming with java A Primer", McGrawHill.
2. Sharanam Shah, " Core Java 8 for Beginners", Shroff Publisher.
3. Naughton&Schildt, "The Complete Reference Java 2", Tata McGraw Hill.
4. Horstmann& Cornell, "Core Java 2" (Vol I &II), Pearson.

Course Outcomes:
On the completion of this course students will be able to understand:
1. The concepts of Java programming
2. The basic terminology used in computer programming and write, compile and debugprograms in JAVA language.
3. The different data types, decision structures, loops, functions to design Java programs.
4. Develop program using the java collection API as well as the java standard class library.
5. Develop Java applets

List of Experiments:
1. Write a program that accepts two numbers from the user and print their sum.
2. Write a program to calculate addition of two number using prototyping of methods.
3. Program to demonstrate function overloading for calculation of average.
4. Program to demonstrating overloaded constructor for calculating box volume.
5. Program to show the detail of students using concept of inheritance.
6. Program to demonstrate package concept.
7. Program to demonstrate implementation of an interface which contains two methods declaration square and cube.
8. Program to demonstrate exception handling in case of division by zero error.
9. Program to demonstrate multithreading.
10. Program to demonstrate JDBC concept using create a GUI based application for student information.
11. Program to display "Hello World" in web browser using applet.
12. Program to add user controls to applets.

13. Write a program to create an application using concept of swing.
14. Program to demonstrate student registration functionality using servlets with session management.