Vertical – 1
Major

Sem. – IV

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344111	Applied Mathematics Thinking-II	02	-	01	02	-	01	03

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344111	Applied Mathematics Thinking-II	20	20	40	60	2	25	--	125

Rationale:

Concepts in Statistics and Operation Research have wide applications in various fields of Engineering. Hence, they are included in the syllabus to help students understand how to apply them in practical applications.

Course Objectives:

The course aims:

1. To study Matrix algebra and its application in engineering problems.
2. To study Contour integrals and expansion of complex valued function in a power series.
3. To study Z-Transforms and Inverse Z-Transforms with its properties.
4. To study the concepts of probability distributions and sampling theory.
5. To study and apply Linear programming Techniques to solve the optimization Problems.
6. To study and apply Non-Linear programming Techniques to solve the optimization problems.

Course Outcomes:

Sr. No.	Course Outcomes	Cognitive levelsof attainment asper Bloom's Taxonomy
On successful completion, of course, learner/student will be able to:		
1	Apply the concepts of eigenvalues and eigen vectors to solve engineering problems.	L1, L2
2	Apply the use of concepts of Complex Integration for evaluating integrals, computing residues & evaluate various contour integrals.	L1, L2
3	Apply the concept of Z- transformation and its inverse in engineering problems.	L1, L2, L3
4	Apply the concept of probability distribution to engineering problems & testing hypothesis of small samples using sampling theory.	L1, L2, L3
5	Apply the concept of Linear Programming to solve the optimization problems.	L2, L3
6	Apply the concept of Non-Linear Programming to solve the optimization problems.	L1, L2, L3

Prerequisite: Applied Mathematics-I, Applied Mathematics-II, Applied Mathematics Thinking-I.

DETAILED SYLLABUS:

Sr. No.	**Name of Module**	**Detailed Content**	**Hours**	**CO Mapping**
I	**Linear Algebra (Theory of Matrices)**	1.1 Characteristic Equation, Eigenvalues and Eigenvectors and properties(without proof). 1.2 Cayley-Hamilton Theorem (without proof), Reductionof higher degree polynomials. 1.3 Similarity of matrices, diagonalizable and non-diagonalizable matrices **Self-learning Topics:** Derogatory and non-derogatory matrices, Functions ofSquare Matrix, Linear Transformations, Quadratic forms.	**05**	**CO1**
II	**Complex Integration**	2.1 Cauchy's theorem, Cauchy's integral formula (without proof). 2.2 Taylor's and Laurent's series (without proof) (upto degree 2 in denominator). 2.3 Definition of Singularity, Zeroes, poles of *f(z)*, Residues, Cauchy'sResidue Theorem (without proof). **Self-learning Topics:** Application of Residue Theorem to evaluate realintegrations.	**07**	**CO2**
III	**Z Transform**	3.1 Definition and Region of convergence. Transform of standard functions: $\{k^n a^k\}$, $\{a^{\lvert k\rvert}\}$, $\{{}^{k+n}C_{\mathrm{n}}\, a^k\}$, $\{c^k \sin(\alpha k + \beta)\}$, $\{c^k \sinh \alpha k\}$, $\{c^k \cosh \alpha k\}$. 3.2 Properties of Z Transform: Change of Scale, Shifting Property,Multiplication, and Division by k, Convolution theorem (without proof). 3.3 Inverse Z transform: Partial Fraction Method, Convolution Method. **Self-learning Topics:** Initial value theorem, Final value theorem, Inverse of Transform by Binomial Expansion.	**07**	**CO3**
IV	**Probability Distribution and Sampling Theory**	4.1 Probability Distribution: Binomial, Poisson and Normal distribution (simple problems in finding probability only to be included). 4.2 Sampling distribution, Test of Hypothesis, Level of Significance, Critical region, One-tailed, and two-tailed test, Degree of freedom. 4.3 Test significance for Large samples. Test the significance of mean and	**08**	**CO4**

		Difference between the means of two large samples. 4.4 Students’ t-distribution (Small sample). Test the significance of mean and Difference between the means of two samples. **Self-learning Topics:** Estimateparameters of a population., Chi-Square Test: Test of goodness of fit and independence of attributes, Contingency table, Yate’s Correction.		
V	**Linear Programming Problems**	5.1 Types of solutions, Standard and Canonical of LPP, Basic and Feasiblesolutions, slack variables, surplus variables. 5.2 Simplex method. 5.3 Duality, Dual of LPP and Dual Simplex Method. **Self-learning Topics:** Artificial variable method, Big M method, Sensitivity Analysis, Two-Phase Simplex Method,Revised Simplex Method.	**06**	**CO5**
VI	**Nonlinear Programming Problems**	6.1 NLPP with one equality constraint (two or three variables) using themethod of Lagrange’s multipliers 6.2 NLPP with one inequality constraint (two variables): Kuhn-Tucker conditions **Self-learning Topics:** Problems with two equality constraints, Problems with two inequality constraints, Unconstrained optimization: One dimensional search method (GoldenSearch method, Newton’s method). Gradient Search method	**06**	**CO6**
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. Operations Research, Hira and Gupta, S. Chand Publication.
2. Linear Algebra, A. R. Vashishta and J. N. Sharma.
3. Fundamentals of Statistics, S. C. Gupta.
4. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa.

References:

1. Advanced Engineering Mathematics, Erwin Kreyszig, John Wiley & Sons.
2. Complex Variables and Applications, Brown and Churchill, McGraw-Hill education.
3. Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill education.
4. Operations Research: An Introduction, Hamdy A Taha, Pearson.
5. Engineering Optimization: Theory and Practice, S.S Rao, Wiley-Blackwell.

Online References:

Sr. No.	**Website Name**
1.	https://www.nptel.ac.in

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

- **Question paper format**

  - Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**

  - **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)

  - A total of **Four questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344112	Operating System	3	–	--	3	–	–	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344112	Operating System	20	20	40	60	2	--	--	100

Course Objectives:

Students will able to learn:

1) To understand the components of Operating System &its functions.
2) To introduce the concept of a process and process management.
3) To understand basic concepts related to Inter-process Communication (IPC) like race condition, mutual exclusion, deadlock, etc. and role of an Operating System in IPC.
4) To understand the concepts and implementation of memory management policies and virtual memory.
5) To understand functions of Operating System for storage management and device management.
6) To study the need and fundamentals of special-purpose operating system with the advent of new emerging technologies.

Course Outcomes:

Upon completion of the course, students will be able to:

Sr. No.	Course Outcomes	Cognitive levels ofattainment as per Bloom's Taxonomy
1	Define operating systems & understand the objective of an OS & its functions.	L1, L2
2	Describe the Process, PCB & compare various process scheduling algorithms.	L1, L2, L3, L4, L5
3	Evaluate the requirement for process synchronization and coordination handled by the operating system.	L2, L3, L4, L5
4	Describe and analyze memory management, its allocation policies, and virtual memory.	L2, L3, L4, L5
5	Analyze and evaluate the services provided by the Operating System for storage management.	L2, L3, L4, L5
6	Compare the functions of various special-purpose Operating Systems.	L1, L2

Prerequisite: Programming language (C & Python).

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Programming Language C; Basic of Hardware i.e. ALU, RAM, ROM, HDD	**02**	**--**
I	**Introduction to Operating Systems**	**Basics of Operating System:** Definition, Types of Operating Systems, OS Structure and operations, Process management, Memory management, storage management. **System Structure**: Operating system services and interface, System calls and types, System boot, Operating System Design and implementation, OS structure, Virtual machines. **Self-learning Topics:** Study of any three different OS, System calls with examples for different OS	**04**	**CO1**
II	**Process Management**	**Processes:** Definition, Process states, Process State transitions, Process Control Block, Context switching, Threads, Mmultithreading, Thread models, Benefits of threads. **Process Scheduling**: Definition, Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time, Scheduling algorithms: Preemptive and Non-preemptive, Thread Scheduling and Multiple Processor Scheduling. **Self-learning Topics:** Performance comparison of Scheduling Algorithms, Selection of Scheduling Algorithms for different situations, Real-time Scheduling.	**06**	**CO2**
III	**Process Synchronization**	**Synchronization**: Inter-process Communication and Synchronization; Race Condition; The Critical Section Problem, Peterson's Solution, synchronization Hardware and semaphores, Producer Consumer Problem; Message passing. **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. **Self-learning Topics:** Study a real time case study for Deadlock detection and recovery.	**09**	**CO3**
IV	**Memory Management**	**Memory Management strategies:** Background, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, Fixed and variable partition, Internal and External fragmentation and Compaction, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation; Virtual Memory basics, Hardware and control structures, Locality of reference, Demand Paging, Page replacement Algorithms, Thrashing. **Self-learning Topics:** Memory Management for any one Operating System, Implementation of Page Replacement Algorithms.	**08**	**CO4**
V	**File Management**	**File system:** File Concept, Access Methods, Directory and DiskStructure, File-System Mounting, File Sharing, Protection; Implementing file System: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery. Overview of Mass Storage Structure; Disk Structure; Disk Scheduling; RAID Structure; Introduction to I/O Systems. **Self-learning Topics:** File System for Linux and Windows, Features of I/O facility for different OS.	**06**	**CO5**

VI	**Special-purpose Operating Systems**	Open-source and Proprietary Operating System; Fundamentals of Distributed Operating System; Network Operating System; Embedded Operating Systems; Cloud and IoT Operating Systems; Real-Time Operating System; Mobile Operating System; Multimedia Operating System; Comparison between Functions of various Special-purpose Operating Systems. **Self-learning Topics:** Case Study on any one Special-purpose Operating Systems.	**04**	**CO6**
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. A. Silberschatz, P. Galvin, G. Gagne, Operating System Concepts, 10th ed., Wiley, 2019.
2. W. Stallings, Operating Systems: Internal and Design Principles, 9th ed., Pearson, 2018.
3. A. Tanenbaum, Modern Operating Systems, Pearson, 4th ed., 2015.

References:

1. N. Chauhan, Principles of Operating Systems, 1st ed., Oxford University Press, 2014.
2. A. Tanenbaum and A. Woodhull, Operating System Design and Implementation, 3rd ed., Pearson.
3. R. Arpaci-Dusseau and A. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, CreateSpace Independent Publishing Platform, 1st ed., 2018.

Online Resources:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106/105/106105214/
2.	https://www.tutorialspoint.com/operating_system/index.htm
3.	https://swayam.gov.in/
4.	https://www.geeksforgeeks.org/operating-systems/

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

➢ **Question paper format**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**

- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)

- A total of **Four questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344113	**Computer Network & Network Design**	**3**	**--**	**--**	**3**	**–**	**–**	**3**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344113	**Computer Network & Network Design**	**20**	**20**	**40**	**60**	**2**	**--**	**--**	**100**

Rationale:

In today's modern world, computer networks serve as the foundation for effective data transfer, facilitating communication through email, messaging, file sharing, video calls, and streaming etc. They also allow devices to share resources such as printers, copiers, and fax machines, leading to significant cost savings. The principles of network design are relevant across various settings, from small businesses to large corporations, highlighting the versatility and importance of this knowledge. As a result, a solid understanding of computer networks is essential for anyone aspiring to work in IT, cybersecurity, or software development.

Course Objectives:

Students will be able to learn:

Sr. No.	Course Objectives
1.	Study the functionalities of each layer of the OSI and TCP/IP models.
2.	Acquire knowledge of different types of transmission media.
3.	Acquire the knowledge of data link layer concepts and their protocols for node-to-node delivery of data.
4.	Analyze the strengths and weaknesses of routing protocols and gain knowledge about IP addressing.
5.	Study the data transportation and session management issues and related protocols used for end-to-end delivery of data.
6.	Gain the knowledge of data presentation techniques used in the presentation layer & client/server model in application layer protocols.

Course Outcomes:

On successful completion, of course, the learner/student will be able to:

Sr. No.	Course Outcomes	Cognitive levels ofattainment as per Bloom's Taxonomy
CO1:	Explain the functionalities of different layers of the OSI & TCP/IP models and compare the models.	L1, L2
CO2:	Categorize the types of transmission media.	L1, L2
CO3:	Explain data link layer concepts, design issues, and protocols.	L1, L2, L3
CO4:	Analyze the network and select an appropriate routing strategy / addressing scheme to design a network for an organization.	L1, L2, L3
CO5:	Describe the mechanisms and related protocols used for end-to-end delivery of data.	L1, L2, L3
CO6:	Implement compression strategies for the application in hand and establish client-server model.	L1, L2, L3

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	• Terminologies of communication	--	--
I	**Introduction to Computer Networks**	• Communication Model • Goals of Computer Communication Networks • Types of Computer Communication Networks • Network Topology • Layered Architecture of Data Network • OSI Reference Model • TCP/IP Reference Model • Internetworking and Network Devices	**04**	**CO1**
II	**Physical Layer**	• Guided and Unguided Transmission Media • Switching: Circuit Switching, Message Switching, Datagram Packet Switching, Virtual-Circuit Packet Switching • Structure of a switch, Space Division Switching, Time Division Switching, Packet Switch	**04**	**CO2**
III	**Data Link Layer**	*Data Link Control:* • Framing • Error Detection & Correction: basic concept, Linear Block Code, CRC code, Checksum (Simple Problems) • Flow Control: Stop-and-Wait Flow Control, Sliding Window Flow Control (Simple Problems) • Error Control: Stop-and-Wait ARQ, Go-Back-N ARQ, Selective Reject ARQ (Simple Problems) *Medium Access Control:* • Scheduled Access (Reservation, Polling, Token Passing), • Random Access (ALOHA, CSMA, CSMA/CD, CSMA/CA) (Simple Problems on throughput) *Link Layer Addressing:* • MAC address and its types • ARP *Ethernet Protocol:* • Standard Ethernet, Fast Ethernet (100 Mbps), Gigabit Ethernet, 10-Gigabit Ethernet **Self-Study Topics:** • HDLC Protocol, IEEE 802.11 Wireless LAN, Bluetooth	**09**	**CO3**
IV	**Network Layer**	*Data Transfer:* • Network Layer Services • IPv4 Addressing (Classful/Classless) (Simple Problems) • Subnetting & Supernetting (Simple Problems)	**10**	**CO4**

		• IPv4 Protocol, ICMP Protocol • IPv6 Addressing & Protocol • Transition from IPv4 to IPv6 *Routing of Packets:* • Routing Algorithms (Distance Vector Routing [Simple Problems], Link State Routing [Simple Problems], Path Vector Routing) • Routing Protocols (RIP, OSPF, BGP) *Network Design:* • Concept of VLAN, VPN • Case study on designing a network for an organization/ college **Self-Study Topics:** • NAT, IGMP, ICMPv6		
V	**Transport Layer & Session Layer**	*Transport Layer:* • Transport Layer Services, Port Number, Socket Address, Flow & Congestion Control at Transport Layer, Connectionless and Connection-Oriented Services • User Datagram Protocol (UDP): User Datagram, UDP Services, UDP Applications (Simple Problems on UDP Header) • Transmission Control Protocol (TCP): TCP Services, TCP Segment Format, TCP Timers (Simple Problems on TCP Header) *Session Layer:* • Session Layer Design Issues • Remote Procedure Call (RPC) Protocol (Handshaking) **Self-Study Topics:** • Congestion control, Quality of Service	**06**	**CO5**
VI	**Presentation Layer & Application Layer**	*Presentation Layer:* • Compression: Basics of compression, Lossless and Lossy Compression • Compression Techniques: Huffman Code, LZW Code, Run Length Code (Simple Problems) • Image Compression: GIF, JPEG *Application Layer:* • Client/Server Paradigm • Standard Applications: WWW, HTTP, FTP, Email, DNS **Self-Study Topics:** • POP, IMAP, SNMP	**06**	**CO6**
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. Behrouz A. Forouzan, Data Communications and Networking with TCPIP Protocol Suite, 6th Edition, McGraw Hill Education, 2022.
2. Andrew S Tanenbaum, Computer Networks -, 6th Edition, Pearson Education, 2022.

References:

1. Behrouz A. Forouzan, TCP/IP Protocol Suite,4th Edition, McGraw Hill Education, 2017.
2. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 8th Edition, Pearson, 2022.
3. Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, 6th Edition, Morgan Kaufmann, 2021.
4. Alberto Leon-Garcia, Indra Widjaja, Communication Networks: Fundamental Concepts and Key Architectures, 2th Edition, McGraw Hill Education, 2017.
5. Stallings William, Data and Computer Communications, 10th Edition, Pearson, 2017.
6. Khalid Sayood, Introduction to Data Compression, 5th Edition, Elsevier, 2019.

Online References:

1. NPTEL Course: Computer Networks and Internet Protocol, by Prof. Soumya Kanti Ghosh, Prof. Sandip Chakraborty, IIT Kharagpur. https://nptel.ac.in/courses/106105183

2. NPTEL Course: Computer Networks, by Prof. Hema A Murthy, IIT Madras. https://nptel.ac.in/courses/106106091

Assessment:
Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

➢ **Question paper format**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Four questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344114	Unix Lab	–	2	–	–	1	--	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344114	Unix Lab	--	--	--	--	--	25	25	50

Lab Objectives:

Students will be able to learn:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To understand architecture and installation of Unix Operating System
2	To learn Unix general purpose commands and programming in Unix editor environment
3	To understand file system management and user management commands in Unix.
4	To understand process management and memory management commands in Unix
5	To learn basic shell scripting.
6	To learn scripting using awk and perl languages.

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, learner/student will be able to:		
1	Understand the architecture and functioning of Unix	L1, L2
2	Identify the Unix general purpose commands	L4
3	Apply Unix commands for system administrative tasks such as file system management and user management.	L3
4	Execute Unix commands for system administrative tasks such as process management and memory management	L4
5	Implement basic shell scripts for different applications.	L3
6	Implement advanced scripts using awk & perl languages and grep, sed, etc. commands for performing various tasks.	L3

Prerequisite: Programming Language C

Hardware & Software Requirements:

Hardware Requirement:	Software requirement:
PC i3 processor and above	Unix, Editor, Bash shell, Bourne shell and C shell

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basic Programming Skills, Concepts of Operating	02	-

		System		
I	Introduction to Unix	Case Study: Brief History of UNIX, Unix Architecture; Installation of Unix Operating System	**03**	LO1
II	Basic Commands	a) Execution of Unix General Purpose Utility Commands like echo, clear, exit, date, time, uptime, cal, cat, tty, man, which, history, id, pwd, whoami, ping, ifconfig, pr, lp, lpr, lpstat, lpq, lprm, cancel, mail, etc. b) Working with Editor Vi/other editor.	**03**	LO2
III	Commands for File System Management and User Management	a) Study of Unix file system (tree structure), file and directory permissions, single and multiuser environment. b) Execution of File System Management Commands like ls, cd, pwd, cat, mkdir, rmdir, rm, cp, mv, chmod, wc, piping and redirection, grep, tr, echo, sort, head, tail, diff, comm, less, more, file, type, wc, split, cmp, tar, find, vim, gzip, bzip2, unzip, locate, etc. c) Execution of User Management Commands like who, whoami, su, sudo, login, logout, exit, passwd, useradd/adduser, usermod, userdel, groupadd, groupmod, groupdel, gpasswd, chown, chage, chgrp, chfn, etc.	**04**	LO3
IV	Commands for Process Management and Memory Management	a) Execution of Process Management Commands like ps, pstree, nice, kill, pkill, killall, xkill, fg, bg, pgrep, renice, etc. b) Execution of Memory Management Commands like free, /proc/meminfo, top, htop, df, du, vmstat, demidecode, sar, pagesize, etc.	**04**	LO4
V	Basic Scripts	a) Study of Shell, Types of Shell, Variables andOperators b) Execute the following Scripts (at least 6): (i) Write a shell script to perform arithmetic operations. (ii) Write a shell script to calculate simple interest. (iii) Write a shell script to determine largest among three integer numbers. (iv) Write a shell script to determine a given year is leap year or not. (v) Write a shell script to print multiplication table of given number using while statement. (vi) Write a shell script to search whether element is present is in the list or not. (vii) Write a shell script to compare two strings. (viii) Write a shell script to read and check if the directory / file exists or not, if not make the directory / file. (ix) Write a shell script to implement menu-driven calculator using case statement. (x) Write a shell script to print following pattern: * * * * * *	**04**	L02, L03, L05

		* * * * (xi) Write a shell script to perform operations on directory like: display name of current directory; display list of directory contents; create another directory, write contents on that and copy it to a suitable location in your home directory; etc.		
VI	Advanced Scripts	a) Execute the following scripts using grep / sed commands: (i) Write a script using grep command to find the number of words character, words and lines in a file. (ii) Write ascriptusing egrep command to display list of specific type of files in the directory. (iii) Write a script using sed command to replace all occurrences of particular word in given a file. (iv) Write a script using sedcommand to print duplicated lines in input. b) Execute the following scripts using awk / perl languages: (i) Write an awk script to print all even numbers in a given range. (ii) Write an awk script to develop a Fibonacci series (take user input for number of terms). (iii) Write a perl script to sort elements of an array. (iv) Write a perl script to check a number is prime or not.	**06**	LO2, L03, L06

Text Books:

1. S. Das, Unix Concepts and Applications, 4th ed., McGraw Hill, 2017.
2. R. Michael, Mastering Unix Shell Scripting, 2nd ed., Wiley, 2008.
3. D. Ambawade, D. Shah, Linux Labs and Open Source Technologies, Dreamtech Press, 2014.

References:

1. Y. Kanetkar, Unix Shell Programming, BPB Publications, 2003.
2. B. Forouzan and R. Gilberg, Unix and Shell Programming, Cengage Learning, 2003.

Assessment :
Term Work: Term work shall consist of at least 10-12 practical’s based on above list. Also Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on Practicals.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344115	Network Design Lab	–	2	–	–	1	--	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344115	Network Design Lab	--	--	--	--	--	25	25	50

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To get familiar with the basic network administration commands
2	To install and configure network simulator and learn basics of TCL scripting.
3	To understand the network simulator environment and visualize a network topology and observe its performance
4	To implement client-server socket programs.
5	To observe and study the traffic flow and the contents of protocol frames.
6	To design and configure a network for an organization

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, the learner/student will be able to:		
1	Execute and evaluate network administration commands and demonstrate their use in different network scenarios	L3, L5
2	Demonstrate the installation and configuration of the network simulator.	L1, L2
3	Demonstrate and measure different network scenarios and their performance behavior.	L1, L2
4	Implement the socket programming for client-server architecture.	L3
5	Analyze the traffic flow of in TCP/IP protocols.	L4
6	Design a network for an organization using a network design tool.	L6

Prerequisite: C /Python programming. Basic commands of Windows and Linux operating systems.

Hardware & Software Requirements:

Hardware Requirement:	Software requirement:
PC i3 processor and above	NS2.34 or higher version, Protocol Analyzer (e.g., Wireshark), C/Java/python

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
I	**Fundamentals of Computer Network**	• To study basic networking commands in windows operating system- Understanding Basic networking Commands: ifconfig ,ip, traceroute, tracepath, ping, netstat, ss, dig, nslookup, route, host, arp, hostname, curl or wget, mtr, whois, tcpdump • Execute and analyze basic networking commands. • To study basic networking commands in Unix/Linux operating system	**02**	**LO1**
II	**Network Simulation Software**	• Installation and configuring of NS-2 simulator-Introduction to Tcl Hello Programming. • Installation and configuring of Cisco Packet Tracer.	**02**	**LO2**
III	**Simulation of Network Topologies and Protocols**	• To implement number of nodes and physical layer configuration using NS2. • To implement the given network topology and transmit data over the shared links using NS2. • Implement distance vector and link state routing protocols in NS2. • To Simulate and study stop and Wait protocol using NS2. • To Simulate Sliding Window protocol using NS2. • To configure and compare different network topologies using Cisco Packet Tracer. • To configure static routes in a network using Cisco Packet Tracer. • Performing dynamic routing in a network using Cisco Packet Tracer. • To perform subnetting/supernetting using Cisco Packet Tracer. • To configure DNS, DHCP, FTP, SMTP server (any one) on Cisco Packet Tracer. • To create a VLAN using Cisco Packet Tracer.	**10**	**LO3**
IV	**Socket Programming**	• To study and implement Socket Programming using TCP. • To study and implement Socket Programming using UDP.	**04**	**LO4**
V	**Protocol Analyzer**	• Install one of the Network protocol analyzer tools like Wireshark, tcpdump, Windump, Microsoft Message Analyzer, Ettercap, Nirsoft SmartSniff etc. • To simulate TCP/IP stack using Network protocol analyzer and analyze the network traffic.	**04**	**LO5**
VI	**Network Design**	• Design a network for an organization using the concepts of Addressing (IP Address Assignment), Naming (DNS) and Routing. • Case Study: Study the network of the institute to identify/understand transmission media, connectors, networking devises, addressing scheme, security features used.	**04**	**LO6**

Text Books:

1. Computer Network Simulation in NS2 Basic Concepts and Protocol Implementation.-Prof Neeraj Bhargava, Pramod Singh Rathore, Dr. Ritu Bhargava, Dr. Abhishek Kumar, First Edition. BPB Publication.
2. Packet analysis with Wire shark, Anish Nath, PACKT publishing
3. TCP/IP Protocol Suite 4th Edition by Behrouz A. Forouzan

References:

1. The Network Simulator ns-2: Documentation: https://www.isi.edu/websites/nsnam/ns/ns-documentation.html
2. Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems by Chris Sanders
3. Cisco Packet Tracer: https://www.netacad.com/cisco-packet-tracer

Assessment :
Term Work: Term work shall consist of at least 10-12 practical's based on above list. Also Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on Practicals.

Vertical – 4
VSEC

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344411	Mini-Project - Programming Paradigm	--	2*+2	--	--	2	--	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344411	Mini-Project - Programming Paradigm	--	--	--	--	--	50	25	75

Lab Objectives:

1. **Understand** the fundamental concepts of different programming paradigms—including Imperative, Declarative, Procedural, Object-Oriented, Functional, Event-Driven, and Logic programming—and **explain** core language design elements such as names, scopes, bindings, type systems, and memory management.
2. **Analyze** the strengths, limitations, and best practices of each programming paradigm, **evaluating** their suitability for solving a diverse range of computational problems.
3. **Apply** the most appropriate programming paradigms to **develop** efficient software solutions, tailoring these solutions to specific problem domains while ensuring performance, modularity, and scalability.
4. **Integrate** object-oriented programming principles, functional programming techniques, and concurrent programming strategies to **design** real-world applications that are robust, maintainable, and efficient.
5. **Implement** logic programming techniques to **solve** declarative problem-solving tasks in areas such as knowledge-based systems, automated reasoning, and artificial intelligence, specifically using languages like Prolog.
6. **Create** scalable and high-performance software architectures by **architecting** solutions that leverage modern design patterns, concurrency models, and advanced programming techniques aligned with real-world needs.

Lab Outcomes:

1. Students will **identify** and **compare** various programming paradigms and their core language design concepts, including names, scopes, bindings, and type systems.
2. Students will **develop and implement** imperative and procedural programming solutions, demonstrating control flow, structured programming, and parameter-passing techniques in languages such as C and Dart.
3. Students will **apply** object-oriented principles, including inheritance, polymorphism, and encapsulation, to **design** software solutions. They will also **analyze** the use of advanced OOP techniques such as interfaces, abstract classes, and exception handling in Java and C++.
4. Students will **understand** functional programming concepts such as pure functions, lambda calculus, and higher-order functions, and **apply** them to design declarative programs in Haskell.
5. Students will **evaluate** logic programming paradigms and **apply** them to solve complex declarative problems, using Prolog to create rules, facts, and queries, with a focus on goal-oriented execution and resolution strategies.

6. **Apply and Create:** Students will **apply** concurrency concepts, including multithreading and synchronization, and **create** event-driven systems using languages like Java and Dart. They will also **develop** scalable solutions to avoid race conditions, deadlocks, and ensure high performance.

Hardware & Software Requirements:

Prerequisite:
DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	**Compilation and interpretation, Focus on overview of compilation steps.**	**1**	–
1	**Introduction to Programming Paradigms and Core Language Design Concepts**	● Introduction to different programming paradigms. **Key Paradigms**: Imperative, Declarative, Procedural, Object-Oriented, Functional, Event-Driven, and Logic programming ● Names, Scopes, and Bindings, Scope Rules, Storage Management. ● Type Systems, Type Checking, Equality Testing and Assignment. **Self Study Topics**: Study of different programming styles in Dart, Python, and Scala	5	CO 1
2	**Imperative and Procedural Paradigm**	● **Imperative Programming:** Core concepts like variables, data types, control flow abstractions (loops, conditionals). ● **Procedural Programming:** Structuring code using procedures, subroutines, and functions. Parameter passing methods - pass by value, pass by address, pass by reference. ● **Structured Programming:** Emphasizing modularity, code readability, and scope management. **Languages**: C, Dart **Self-Study Topics:** Exploring programming implementations to understand the use of Generic subroutines, Modules, Coroutines and Event handling	4	CO 2

3	**Object-Oriented Paradigm**	● **Core OOP Concepts**: Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic binding and Object Lifecycle. ● **Advanced OOP Techniques**: Interfaces, Abstract Classes. ● **Exception Handling** **Languages**: Java, C++ **Self-Study Topics:** Analyzing and designing applications using OOP principles.	5	CO 3
4	**Functional Programming Paradigm**	● Introduction to Lambda Calculus ● **Functional Programming features-** Pure Functions, functions as first class members, Curried Functions, Basic and Composite Types and Type Classes ● Pattern Matching, Guard Expressions, Evaluation Order, Higher Order Functions **Language: Haskell** **Self-Study Topics:** Identifying functional features in modern programming languages like Dart, Python.	4	CO 4
5	**Logic Programming Paradigm**	● **Logic Programming Basics:** Declarative programming, facts, rules, and queries. ● **Prolog Basics:** Syntax, unification, backtracking, and logical inference. ● **Resolution and Proof Search:** Goal-oriented execution and resolution strategies. **Languages**: **Prolog** **Self-Learning Topics:** Use of logic programming for expert systems, automated reasoning, and AI.	4	CO 5
6	**Concurrent and Event-Driven Paradigms**	● **Concurrency Concepts**: Multithreading, parallelism, synchronization, and avoiding race conditions and deadlocks. ● **Event-Driven Programming**: Event loops, callbacks, asynchronous programming, and handling asynchronous events.	6	CO 6

		Languages: Java (Concurrency model), Dart (Limited to study of event driven programming pattern)		

Textbooks

1	Scott M L, Programming Language Pragmatics, 3rd Edn., Morgan Kaufmann Publishers, 2009
2	Graham Hutton, Programming in Haskell, 2nd Edition, Cambridge University Press, 2016
3	Ravi Sethi,, Programming Languages: Concepts and Constructs; 2nd Edition, Pearson Education Asia, 1996.
4	Jonathan Sande, Kodeco Tutorial Team, Dart Apprentice: Beyond the Basics (First Edition): Object-Oriented Programming, Concurrency & More: 1st edition, Kodeco Incorporated, 2022, 2022

Reference Books

1	Programming Languages: Design and Implementation (4th Edition), by Terrence W. Pratt, Marvin V. Zelkowitz, Pearson, 2000
2	Rajkumar Buyya, Object-oriented Programming with Java: Essentials and Applications, Tata McGraw Hill Education Private Limited
3	Max Bramer, Logic Programming with Prolog, Springer ISBN-13: 978-1852-33938-8

Online Resources

Sr	**Module**	**Free Online Courses**
1	Module I	https://see.stanford.edu/Course/CS107 https://ocw.mit.edu/collections/introductory-programming/ https://see.stanford.edu/Course/CS106A Principles of programming Languages, https://nptel.ac.in/courses/106102067
2	Module II	
3	Module III	
4	Module IV	Learn You Haskell For Great Good, https://learnyouahaskell.com/chapters https://www.coursera.org/learn/scala-functional-programming
5	Module V	Online Prolog Learning Resources, https://swish.swi-prolog.org/example/examples.swinb https://www.coursera.org/learn/logic-introduction https://www.coursera.org/learn/an-introduction-to-logic-in-computer-science https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005/resources/8a-logic-programming-part-1/
6	Module VI	Book Companion Resource : Dart Apprentice: Beyond the Basics (First Edition): Object-Oriented Programming, Concurrency & More https://github.com/kodecocodes/dabb-materials/tree/editions/1.0 https://see.stanford.edu/Course/CS107 https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005/resources/8a-logic-programming-part-1/

Guidelines for Labs Capstone Mini-Project:

Suggested List of Experiments

Sr. No.	**Module**	**Detailed Content**	**Hours**	**LO Mapping**
0	Prerequisite	Demonstrate Compilation and interpretation stages to students for C, C++, JAVA along with how to debug the code.	02	–
I	Object Oriented Programming	**At least Two** Programming Implementations Preferably in C++ to demonstrate concepts like - Abstraction & Encapsulation, Initialization and Finalization, Inheritance, Polymorphism and Dynamic Binding.	06	LO 1
II	Run Time Program Management	**At least Two** Programs to understand Exception handling and Garbage collection, preferably in JAVA. Students should understand checked and unchecked exceptions as well as using multiple catch blocks	02	LO 2
III	Concurrent Programming	**At least Two** Program preferably in Java/C++ to demonstrate the Thread management and Synchronization	02	LO 3
IV	Functional Programming	**Tutorial on** Introduction to Haskell programming environment and Basic operators, types, prelude library functions, list and tuples in Haskell **At least Four** Haskell Programs to demonstrate Functional Programming Concepts. Sample Programs but not limited to: ● Implement a safetail function that behaves in the same way as tail, except that safetail maps the empty list to the empty list, whereas tail gives an error in this case. Define safetail using: (a) a conditional expression; (b) guarded equations; (c) pattern matching. ● Simple List Comprehension ● Higher-Order Functions ● Write a recursive function to multiply two natural numbers that use a predefined add function. ● Haskell code to represent infinite list e.g. fibonacci series	06	LO 4

		● Implement simple Calculator Students should clearly understand the syntax and the execution of the Functional Implementation using Haskell.		
V	Logic Programming	**Tutorial** on working of SWI Prolog Environment and basic understanding of facts and rules **Implement at least 2** Prolog programs to understand declarative programming concepts. The programs must be based on creating a Knowledge Base having multiple facts and rules. Students must be able to learn to query it and understand query execution using the backward chaining.	06	LO 5
VI	Concurrency and Parallelism through Dart Programming	Implement any 3 sample programs in Dart based on learning implementation of concurrency, Streams and Isolates	06	LO 6

Guidelines for Capstone Mini-Project

- Students shall form a group of 3 to 4 students, and a group shall not be allowed less than three or more than four students, as it is a group activity.
- Students should do the survey and identify needs, which shall be converted into a problem statement for the mini project in consultation with the faculty supervisor/head of department/internal committee of faculty.
- Students' hall submits an implementation plan in the form of a Gantt/PERT/CPM chart, which will cover the weekly activity of a mini-project.
- A log book to be prepared by each group, wherein the group can record weekly work progress, and the guide/supervisor can verify and record notes/comments.
- Faculty supervisor may give inputs to students during the mini project activity; however, focus shall be on self-learning.
- Students in a group shall understand the problem effectively, propose multiple solutions, and select the best possible solution in consultation with the guide/ supervisor.
- Students shall convert the best solution into a working model using various components of their domain areas and demonstrate.
- The solution to be validated with proper justification and a report to be compiled in a standard format of the University of Mumbai.

Guidelines for Assessment of Capstone Mini-Project:

Term Work

- The review/ progress monitoring committee shall be constituted by the head of departments of each institute. The progress of the mini project to be evaluated on continuous basis, minimum two reviews in each semester.
- In continuous assessment, focus shall also be on each individual student, assessment based on the individual's contribution in group activity, their understanding, and response to questions.
- Distribution of Term work marks for both semesters shall be as below;
  - Marks awarded by guide/supervisor based on above practical list and assignment completed by the students 10

- Marks awarded by guide/supervisor for Capstone Project Completion: :10
- Quality of Project report & Project Review 05

Review/progress monitoring committee may consider following points for assessment based of the semester project as mentioned in general guidelines.

- In this case in students' group shall complete project in all aspects including,
  - Identification of need/problem
  - Proposed final solution
  - Procurement of components/systems
  - Building prototype and testing
- Two reviews will be conducted for continuous assessment,
  - First shall be for the finalisation of problem and proposed solution
  - Second shall be for the implementation, testing and validation of solution.

Assessment criteria of Capstone Mini-Project.

Capstone Mini-Project shall be assessed based on following criteria;

1. Quality of survey/ need identification
2. Clarity of Problem definition based on need.
3. Innovativeness in solutions/ Novelty in solutions.
4. Feasibility of proposed problem solutions and selection of best solution
5. Collection of Dataset.
6. Cost effectiveness
7. Societal impact
8. Innovativeness
9. Cost effectiveness and Societal impact
10. Full functioning of working model as per stated requirements
11. Effective use of skill sets
12. Effective use of standard engineering norms
13. Contribution of an individual's as member or leader
14. Clarity in written and oral communication

Guidelines for Assessment of Capstone Mini-Project Practical/Oral Examination:

- Report should be prepared as per the guidelines issued by the University of Mumbai.
- Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organisations having experience of more than five years approved by head of Institution.
- Students shall be motivated to publish a paper based on the work in Conferences/students competitions.

List of Sample Capstone Mini-Project :

1. Object-Oriented Programming (OOP)

a. Library Management System

- Use classes for Books, Users, Transactions.
- Implement inheritance for different types of users (students, staff).
- Optional: Add GUI using Tkinter or JavaFX.

b. Online Food Ordering System (Console-based)

- Classes: Customer, MenuItem, Order, Restaurant
- OOP features like encapsulation, polymorphism.

c. Expense Tracker App (Java or Python)

- Use OOP to model Users, Transactions, and Categories.

2. Functional Programming

a. Weather Data Analyzer (Haskell / Python with functools)

- Read weather data and apply pure functions for analysis.
- Avoid mutable data.
- Use map, reduce, filter extensively.

b. To-Do List Manager (Scala or Elixir)

- Immutable data structures
- Recursion instead of loops
- No side effects in logic layer.

3. Procedural Programming

a. Student Grade Calculator (C / Python)

- Use functions and data structures to store and compute data.
- No OOP involved.

b. ATM Simulator

- Menu-based system using functions only (no classes).
- PIN check, balance enquiry, cash withdrawal, etc.

4. Logic Programming

a. Family Tree Generator (Prolog)

- Define relations like parent, sibling, cousin using rules.
- Query relationships dynamically.

b. Sudoku Solver (Prolog)

- Define rules and constraints to solve 9x9 Sudoku puzzle.

5. Multi-Paradigm Approach

a. Chatbot using Python (OOP + Functional)

- OOP for structuring code (User, Bot, ChatSession).
- Functional for processing input (filters, parsers).

b. File Organizer Script

- Script to organize files into folders based on type.
- Use procedural logic + OOP for file operations.

6. Event-Driven Programming

a. GUI Calculator (Python Tkinter / Java Swing)

- Event listeners for button presses.
- Follow MVC pattern optionally.

b. Traffic Light Simulation (JavaScript + HTML/CSS)

- Events for light changes
- Use state variables to track behavior

Assessment :

Term Work: Term Work shall consist of list of all practicals' based on the above list. Also, the Term work Journal must include at least 2 assignments and Mini-Project Report.

Term Work Marks: 50 Marks (Total marks) = 10 Marks (Experiment) + 5 Marks (Assignments) + 30 Marks (Capstone Mini- Project with full prototype/ product demo, testing, validation and Report) + 5 Marks (Attendance).

Oral Exam: An Oral exam will be held based on the Capstone Mini-Project.

Sem. – IV

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2994511	Business Model Development	--	2*+2	-	--	2*+2	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I+IAT-II					
2994511	Business Model Development	--	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Lab Objectives:

1. To introduce a learner to entrepreneurship and its role in economic development.
2. To familiarize a learner with the start-up ecosystem and government initiatives in India.
3. To explain the process of starting a business.
4. To familiarize a learner with the building blocks of a business.
5. To teach a learner to plan their own business with the help of Business Model Canvas.
6. To teach a learner to have financial plan for a business model.

Lab Outcomes:

The learner will be able to:

1. Discuss the role of entrepreneurship in the economic development of a nation and describe the process of starting a business.
2. Describe start-up ecosystems in Indian and global context.
3. Identify different types of business models.
4. Identify customer segments, channels and customer relationship components for a particular business.
5. Identify key activities, key partners and key resources for a particular business.
6. Develop a financial plan for a business with the help of cost structure and revenue model.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	**Prerequisite**	Basic Design Thinking principles	**01**	**--**
I	**Introduction to Entrepreneurship**	**Introduction to Entrepreneurship**: Definition, the role of entrepreneurship in the economic development, the entrepreneurial process, Women entrepreneurs, Corporate entrepreneurship, Entrepreneurial mindset **Self-learning Topics:** Case studies: Henry Ford https://www.thehenryford.org/docs/default-source/default-document-library/default-document-library/henryfordandinnovation.pdf?sfvrsn=0 The Tatas: How a Family Built a Business and a Nation by Girish Kuber, April 2019, Harper Business	**04**	**L1, L2**
II	**Entrepreneurship Development**	**Entrepreneurship Development**: Types of business ownerships: Proprietorship, Public and Private Companies, Co-operative businesses, Micro, Small and Medium Enterprises (MSME): Definition and role of MSMEs in economic	**05**	**L2, L3, L4**

		development		
III	**Start-up financing**	**Start-up financing:** Cost and revenue models, Sources of start-up fundings: Angel investors, Venture capitalists, Crowd funding, Government schemes for start-up funding **Self-learning Topics:** Successful business pitching	**04**	**L2, L3, L4, L5**
IV	**Intellectual Property Rights (IPR)**	**Intellectual Property Rights (IPR):** Types of IPR: Patents, trademarks and copyrights, Patent search and analysis, Strategies for IPR protection, Ethics in technology and innovation	**04**	**L2, L3, L4**
V	**Business Model Development**	**Business Model Development:** Types of business models, Value proposition, Customer segments, Customer relationships, Channels, Key partners, Key activities, Key resources, Prototyping and MVP **Self-learning Topics:** The Art of the Start 2.0: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything by Guy Kawasaki	**04**	**L3, L4, L5, L6**
VI	**Digital Business Management**	**Digital Business Management:** Digital Business models (Subscription, Freemium *etc*), Digital marketing: Search Engine Optimization (SEO), Search Engine Marketing (SEM), Social media and influencer marketing, Disruption and innovation in digital business **Self-learning Topics:** Case study: Airbnb https://www.prismetric.com/airbnb-business-m	**04**	**L2, L3**

Textbooks:

1. Entrepreneurship: David A. Kirby, McGraw Hill, 2002
2. Harvard Business Review: Entrepreneurs Handbook, HBR Press, 2018
3. Business Model Generation; Alexander Ostlewalder and Yves Pigneur, Strategyzer, 2010
4. E- Business & E– Commerce Management: Strategy, Implementation, Practice – Dave Chaffey, Pearson Education

Reference books:

1. Entrepreneurship: New venture creation by David Holt, Prentice Hall of India Pvt. Ltd.
2. E- Business & E– Commerce Management: Strategy, Implementation, Practice – Dave Chaffey, Pearson Education

Online Resources:

Sr. No.	Website Name
3.	Entrepreneurship by Prof. C Bhaktavatsala Rao https://onlinecourses.nptel.ac.in/noc20_mg35/preview
4.	Innovation, Business Models and Entrepreneurship by Prof. Rajat Agrawal, Prof. Vinay Sharma https://onlinecourses.nptel.ac.in/noc21_mg63/preview
3.	Sarasvathy's principles for effectuation https://innovationenglish.sites.ku.dk/model/sarasvathy-effectuation/

List of Experiments.

The lab activities are to be conducted in a group. One group can be formed with 4-5 students. A group has to develop a Business Model Canvas and a digital prototype (Web App/ mobile app). Weekly activities are to be conducted as follows:

Sr No	Lab activities	Hrs
01	Problem identification (Pain points, Market survey)	2
02	Design a digital solution for the problem (Ideation techniques)	2
03	Preparing a business model canvas: Value proposition, Key partners, Key resources, Key activities	2
04	Preparing a business model canvas: Customer segment, Customer relationships and channels	2
05	Preparing a business model canvas: Cost and Revenue structure	2
06	Prototype development: Low fidelity	2
07	Prototype development: Customer feedback	2
08	Prototype development: High fidelity	2
09	Presentation of high-fidelity prototype	2

Sr No	List of Assignments / Tutorials	Hrs
01	Presentation on case study of a failed business model	2
02	Presentation on case study of a woman entrepreneur	2

Assessment:

Term Work: Term Work shall consist of 09 lab activities based on the above list. Also, Term work journal must include any 2 assignments from the above list.

Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 10 Marks (Assignments) + 5 Marks (Attendance)+10 Marks (Report).

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2994512	Design Thinking	--	2*+2	-	--	2*+2	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I+IAT-II					
2994512	Design Thinking	--	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Lab Objectives:

1. To introduce a learner to the principles of Design Thinking.
2. To familiarize a learner with the process (stages) of Design Thinking.
3. To introduce various design thinking tools.
4. Study of the techniques for generation of solutions for a problem.
5. To expose a learner to various case studies of Design Thinking.
6. Create and test a prototype.

Lab Outcomes:

Students will be able to …

1. Compare traditional approach to problem solving with the Design Thinking approach and discuss the principles of Design Thinking
2. Define a user persona using empathy techniques
3. Frame a problem statement using various Design Thinking tools
4. Use ideation techniques to generate a pool of solutions for a problem
5. Create prototypes using different techniques
6. Test the prototypes and gather feedback for refining the prototype

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	No perquisites	-	-
I	**Introduction to Design Thinking**	**Introduction to Design Thinking:** Definition, Comparison of Design Thinking and traditional problem-solving approach, Need for Design Thinking approach, Key tenets of Design Thinking, 5 stages of Design Thinking (Empathize, Define, Ideate, Prototype, Test) **Self-learning Topics:** Design thinking case studies from various domains https://www.design-thinking-association.org/explore-design-thinking-topics/external-links/design-thinking-case-study-index	**05**	**L1, L2**
II	**Empathy**	**Empathy:** Foundation of empathy, Purpose of empathy, Observation for empathy, User observation technique, Creation of empathy map	**05**	**L2, L3**

		Self-learning Topics: Creation of empathy maps https://www.interaction-design.org/literature/topics/empathy-mapping		
III	**Define**	**Define:** Significance of defining a problem, Rules of prioritizing problem solving, Conditions for robust problem framing, Problem statement and POV **Self-learning Topics:** Creating a Persona – A step-by-step guide with tips and examples https://uxpressia.com/blog/how-to-create-persona-guide-examples	**05**	**L2, L3**
IV	**Ideate**	**Ideate:** What is ideation? Need for ideation, Ideation techniques, Guidelines for ideation: Multi-disciplinary approach, Imitating with grace, Breaking patterns, Challenging assumptions, Looking across value chain, Looking beyond recommendation, Techniques for ideation: Brainstorming, Mind mapping **Self-learning Topics:** How To Run an Effective Ideation Workshop: A Step-By-Step Guide https://uxplanet.org/how-to-run-an-effective-ideation-workshop-a-step-by-step-guide-d520e41b1b96	**05**	**L3**
V	**Prototype**	**Prototype:** Low and high-fidelity prototypes, Paper prototype, Story board prototype, Scenario prototype	**03**	**L6**
VI	**Test**	**Test:** 5 guidelines of conducting test, The end goals of test: Desirability, Feasibility and Viability, Usability testing	**03**	**L4, L5**

Textbooks:

1. Design Your Thinking: The Mindsets, Toolsets, and Skill Sets for Creative Problem-solving, Pavan Soni, Penguin Random House India Private Limited
2. Design Thinking: Methodology Book, Emrah Yayichi, 2016
3. Handbook of Design Thinking: Christian Mueller-Roterberg, 2018

Reference books:

1. Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School, Idris Mootee, Wiley, 2013
2. Change by Design, Tim Brown, Harper Business, 2009

Online Resources:

Sr. No.	**Website Name**
5.	Design Thinking and Innovation by Ravi Poovaiah https://onlinecourses.swayam2.ac.in/aic23_ge17/preview
6.	Introduction to Design Thinking by Dr. Rajeshwari Patil, Dr. Manisha Shukla, Dr.

	Deepali Raheja, Dr. Mansi Kapoor https://onlinecourses.swayam2.ac.in/imb24_mg37/preview
3.	Usability Testing https://www.interaction-design.org/literature/topics/usability-testing

List of Experiments.

The experiments are to be performed in groups. A practical batch may be divided into groups of 4-5 students.

Sr No	List of Experiments	Hrs
01	Customer Journey Mapping: Visualize the steps users take to interact with a product or service. Map out the customer journey from discovering a product to making a purchase and using the product. Identify pain points and opportunities for improvement.	2
02	Stakeholder mapping: Identify all relevant stakeholders in a project. Create a stakeholder map, categorizing stakeholders based on their influence and interest. Include management of relationships with key stakeholders.	2
03	"How Might We" Problem Framing: Transform user insights into actionable problem statements. After empathizing with users, turn challenges into "How Might We" statements that define the problem without prescribing a solution.	2
04	Brainstorming Session: Generate a pool of ideas in a creative, non-judgmental environment. Using ideation techniques like mind mapping and brainwriting, students brainstorm as many solutions as possible to their "How Might We" problem statements.	2
05	Affinity Diagramming: Organize group ideas to find patterns and insights. After brainstorming, students will categorize their ideas into themes by placing sticky notes on a wall and moving them into groups based on similarities.	2
06	Rapid Prototyping: Create quick, low-fidelity versions of solutions. Use materials like paper, cardboard, and markers to build a prototype of their solution within 30 minutes. The focus is on speed and functionality, not aesthetics.	2
07	Wireframing: Create a visual guide for digital interfaces for mobile app / web app for the problems identified in earlier lab sessions. Students will sketch wireframes of the user interface for their product or service. Use tools like Balsamiq or paper and pen for low-fidelity wireframes.	2
08	Role-Playing: Walk through a prototype from the user's perspective. Students act as both users and designers, role-playing scenarios where they interact with their prototype (Developed in earlier lab sessions). Gather feedback from participants on how to improve the experience.	2
09	Usability Testing: Evaluation of the effectiveness and user-friendliness of a prototype (developed in earlier lab sessions). Students will have peers or target users test their prototypes, observe how they interact with it, and collect feedback on any issues or improvements needed.	2
10	Feedback Loop and Iteration: Refine solutions based on user feedback. After usability testing, students will refine their prototypes. Document changes made based on feedback and discuss how continuous iteration improves the design.	2

Sr No	List of Assignments (Any two)	Hrs
01	Create an empathy map for a target user group. Break them into four sections: *Says, Thinks, Feels, and Does*. Interview users or research their experiences to	3

	fill in the map.	
02	Based on research, students will create user personas including demographic details, motivations, pain points, and goals. Each group will present their persona to the class.	3
03	Consider 3 examples of real-life products which have good design and bad design. Write down reasons why do you think they are good or bad designs. May take user survey to support your work.	3
04	Study any open-source design thinking tool and write a brief report about it.	3

Assessment:

Term Work: Term Work shall consist of 08 to 10 lab activities based on the above list. Also, Term work journal must include any 2 to 4 assignments from the above list.

Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 10 Marks (Assignments) + 5 Marks (Attendance)+ 10 Marks (Report).