



**Government of Karnataka
Department of Technical Education**

C-25 Diploma Curriculum Information Science and Engineering

Scheme of Studies
(Effect from the AY 2025-26)



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

I Semester Scheme of Studies - Diploma in Information Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	SC	25SC11I	Engineering Mathematics-I	4	0	4	8	6	50	20	50	20	-	-	100
2	CS	25CS01I	IT Skills	3	0	4	7	5	50	20	-	-	50	20	100
3	EE	25EE01I	Fundamentals of Electrical & Electronics Engineering.	3	0	4	7	5	50	20	-	-	50	20	100
4	CS	25CS11I	Basics of Digital Logic and Computer Organization	4	0	4	8	6	50	20	50	20	-	-	100
Audit Course															
5	IS	25IS11T	Environmental Sustainability	2	0	0	2	2	50	20	-	-	-	-	50
6	Personality Development		NCC/NSS/YOGA/SPORTS...	Students are expected to engage in any one of these activities from 1 st semester to 6 th semester(No Credits)											
Total				16	0	16	32	24	250	-	100	-	100	-	450



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

II Semester Scheme of Studies - Diploma in Information Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	SC	25SC21I	Engineering Mathematics-II	4	0	4	8	6	50	20	50	20	-	-	100
2	ENG	25EG01I	Essential English Communication	4	0	4	8	6	50	20	-	-	50	20	100
3	ME	25ME02I	Computer Aided Engineering. Graphics	3	0	4	7	5	50	20	-	-	50	20	100
4	CS	25CS21I	Thinking Programming with Python	4	0	4	8	6	50	20	50	20	-	-	100
Audit Course															
5	IS	25IS21T	Indian Constitution	2	0	0	2	2	50	20	-	-	-	-	50
6	Personality Development		NCC/NSS/YOGA/SPORTS...	Students are expected to engage in any one of these activities from 1 st semester to 6 th semester (No Credits)											
Total				17	0	16	33	25	250	-	100	-	100	-	450



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

III Semester Scheme of Studies - Diploma in Information Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	IS	25CS31I	Database Concepts & Technologies	4	0	4	8	6	50	20	50	20	-	-	100
2	IS	25CS32I	Computer Networks	4	0	4	8	6	50	20	50	20	-	-	100
3	IS	25IS33I	Data Structures with Python	3	0	4	7	5	50	20	-	-	50	20	100
4	IS	25IS34I	System and Network Administration	3	0	4	7	5	50	20	-	-	50	20	100
Audit Course															
5	KAN	25KA31T	Kannada –I (ಸಾಹಿತ್ಯ ಸಿಂಚನ-1/ಬಳಕೆ ಕನ್ನಡ-1)	2	0	0	2	2	50	20	-	-	-	-	50
Total				16	0	16	32	24	250	-	100	-	100	-	450



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Curriculum Structure

IV Semester Scheme of Studies - Diploma in Information Science and Engineering

Sl. No.	Teaching Department	Course Code	Course Name	Hours per week			Total Contact Hours/week	Credits	CIE Marks		Theory SEE Marks		Practice SEE Marks		Total Marks
				L	T	P			Max	Min	Max	Min	Max	Min	
Integrated Courses															
1	IS	25CS41I	Operating Systems and Administration	4	0	4	8	6	50	20	50	20	-	-	100
2	IS	25CS42I	Object Oriented Programming with Java	4	0	4	8	6	50	20	50	20	-	-	100
3	IS	25IS43I	Data Analysis and Algorithm Design	3	0	4	7	5	50	20	-	-	50	20	100
4	IS	25IS44I	Web Development	3	0	4	7	5	50	20	-	-	50	20	100
Audit Course															
5	KAN	25KA41T	Kannada –II (ಸಾಹಿತ್ಯ ಸಿಂಚನ-II/ಬಳಕೆ ಕನ್ನಡ-II)	2	0	0	2	2	50	20	-	-	-	-	50
Total				16	0	16	32	24	250	-	100	-	100	-	450

SEMESTER 1



Government of Karnataka

DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	1
Course Name	IT Skills	Type of Course	Integrated
Course Code	25CS01I	Contact Hours	7 per week
Teaching Scheme	3: 0:4	Credits	5
CIE Marks	50	SEE Marks	50 (Practice)

1. Rationale: This content equips learners with essential IT skills needed to thrive in a technology-driven world by fostering digital literacy and problem-solving abilities. Through logical thinking and block coding, students develop computational thinking crucial for solving real-world challenges. Emphasizing cyber safety and security ensures learners adopt responsible and ethical digital practices. Incorporating multimedia design, AI, cloud, and IoT tools prepares students to innovate and leverage emerging technologies. This holistic approach bridges foundational IT skills with advanced applications, enabling learners to adapt and excel in diverse fields.

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

CO-01	Use computer, Internet.
CO-02	Develop problem-solving skills by applying logical thinking and block coding techniques.
CO-03	Develop engaging multimedia presentations.
CO-04	Implement best practices of cyber safety and security.
CO-05	Use Free AI, cloud and IoT services to solve computing problems.

3. Course Content

Week	CO	PO	Lecture(3HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1	Introduction to Computers - Definition and basic understanding of a computer. - Generations of Computers - Types of Computers - Block Diagram of a Digital Computer - Functional Units of the CPU: - ALU - Control Unit Memory Systems: Types of Memory and Their Usage: Primary Memory: RAM(SRAM, DRAM). Cache Memory: Levels of cache (L1, L2, L3) and their purpose in speeding up data access.	1. Identify the parts of a computer system. 2. Create a file, folder and find details of the Operating system and hardware being used. 3. Connect components and install application software such as scratch. 4. List out important factors/criteria to be considered while buying a computer: a. Purpose – Professional, Gaming, General use, Portability b. Budget c. Processor – Brand, Cores and threads, clock speed d. Memory (RAM) – Minimum required and type e. Storage – Type(HDD, SSD), Capacity

			Secondary Memory: ROM (Read-Only Memory), Hard Disk Drives (HDD) and Solid State Drives (SSD) Input/output Systems: Types of I/O Devices, Cables and Connectors Software: System software (e.g., operating systems) vs. application software (e.g., word processors, browsers).	f. Display – Size, Resolution, Refresh rate g. Ports and connectivity: USB, HDMI, Ethernet etc. h. Graphics – Integrated/ Dedicated i. Operating system – Windows, Linux, Mac j. Battery life in case of a laptop k. Build quality and design, Brand warranty etc.
2	1	1,4	Internet skills: What is Computer Networks? Types of Networks. Physical and Logical address, Protocols, Key Devices in a Network (Router, Switch, Modem, Access Point) What is the Internet, Common Applications of the Internet? Browsers, Web Server, Client–Server Model, URL, Search Engine, Domain name and domain name system, websites. Personal website, website hosting.	1. Create Email account such as Gmail and Explore various services offered in Google accounts such as a. Gmail b. docs c. Sheets d. forms etc. 2. Download any application software required for your learning (Such as internet browser) and install it on your machine. 3. Design, develop and host a personal website using any free platform such as wix.com or worldpress.com 4. Manual and Automatic address assignment (Windows and Android) a) IPv4 address b) Subnet mask c) DNS and Default gateway 5. Test Internet speed
3	4	1,4,5,7	Cybersecurity What is Cybersecurity? Confidentiality, Integrity, Availability. Brief awareness of cyber safety measures, Data, Types of Data (Personal, Organizational) Protecting your personal data and devices: Device sync Online identity - Where is your data? Terms of service while signing up for a web application and sharing your data. Home WiFi Security, Public Wi-Fi risks. Basic security issues in your mobile phones and personal computers, Importance of privacy and Password policy (Best practices). Two factor authentication.	1. Demonstrate Two-factor authentication 2. Browser settings 3. Android App permissions and settings. 4. Firewall concepts and settings. 5. Setup Wireless network security.
4	4	1,4,	Security Attacks (Demonstration)	1. Installation of Antivirus software

		5,7	Malware and its types, Phishing, Social Engineering, DoS attack, On-path attack, Man in the middle attack, Eavesdropping, Spamming, Identity theft, Password attacks, SEO Poisoning Etc. Internet Safety Anti-virus software, HTTP vs HTTPS, Firewall, Cookies. Staying Safe from Online Predators, Cyberbullying and Cyber harassment, Using Social Networks Safely.	2. Installing and Updating Browser Plug-ins. 3. Privacy settings in social media accounts. Security audit (Antivirus, strong password, firewall, app settings and permissions, unwanted apps) of all your personal smart computer, college computer and smart phone.
5	2	1,2,3,7	Introduction to Problem Solving Steps for Problem Solving Algorithms: What is Algorithm, How and why to design Algorithm. Sequencing, Selection, iteration.	1. Write an algorithm for programmable problems - Add/subtract two numbers - Find the largest/smallest of 3 numbers - Find the square of a number - Calculate and print the sum of 'N' numbers - check whether a number is odd or even. 2. Write Algorithm to accept numbers till the user enters 0 and then find their average. 3. Write Algorithm to draw following pattern on screen <pre> * * * * * * * * * </pre>
6	2	1,2,3,7	Flowcharts: What is Flow Chart, How and why to design Flow Chart. Sequencing, Selection, iteration.	Write flowchart for programmable problems - Add/subtract two numbers - Find the largest/smallest of 3 numbers - Find the square of a number - Calculate and print the sum of 'N' numbers - check whether a number is odd or even. 2. Write Algorithm and flowchart to accept numbers till the user enters 0 and then find their average. 3. Write Algorithm and Flowchart to draw following pattern on screen <pre> * * * * * * * * * </pre>

				***** ***** *
7	2	1,2, 3,7	Block coding What is block coding? How does block-based coding work? Introduction block coding platforms (MIT scratch or Edublock) Different blocks and their uses Motion Looks Sound Events	1. Explore different code blocks 2. Design and implement a scratch project to move a object 10 steps and say hello. 3. Design and implement scratch project with 2 objects interacting Teacher can use any free visual block coding tools for execution, some examples are a. https://scratch.mit.edu/ b. https:// edublocks.org c. https://mblock.cc
8	2	1,2, 3,7	Block coding Different blocks and their uses Control Sensing Operators Variables	1. Explore different code blocks 2. Design and implement program to check largest of 3 numbers. 3. Design and implement program calculate sum of n even numbers 4. Design a simple game using the MIT scratch tool. 5. Convert block code to python code Teacher can use any free visual block coding tools for execution, some examples are a. https://scratch.mit.edu/ b. https:// edublocks.org c. https://mblock.cc
9	5, 1	2,4, 5	Cloud Computing: - Definition, applications, and real world examples. - Cloud service models (SaaS, PaaS, IaaS) - Deployment models (Public, Private, Hybrid, Community) with examples. Cloud Storage and Online Collaboration - Cloud Storage: Google Drive, OneDrive, Dropbox - Basics of Online Collaboration Tools: Google Docs, Sheets, and Slides - File Sharing and Permissions: Securing Shared Data Web-Based Applications: Online Forms and Surveys	- File management (upload your documents, class notes, study materials), Organizing, and Sharing Files Using Google Drive - Real-Time Collaboration on Google Docs(Edit a document such as your lab observation in Google docs, name the versions and demonstrate.) - Creating and Managing a Shared Task List Using Google Sheets - Creating a Basic Online Survey(feedback) Using Google Forms
10	5, 1	2,4, 5	What is Internet of Things, IoT Examples. How Does IoT works : Sensors, Actuator in IoT, Control unit/data processing. Applications of IoT. Examples of IoT in everyday life: Smart	1. Create a simple visual block code to blink LED in Arduino board using visual block code, upload code to Arduino board and demonstrate. 2. Create a Traffic signal controller with 3

			home, smart lighting, Building information system, Automation in Manufacturing, IoT in automobiles etc. The Future of IoT.	LED (RED, YELLOW and GREEN), upload code to Arduino board and demonstrate. Note : Students and Teachers to use visual block code platform such as a. https://www.tinkercad.com/ b. https://mblock.cc for building IoT application and demonstration.
11	3	4,7	Multimedia and Presentation: Multimedia Building blocks Text, Audio, Image, Animation, Video Resolution, Size, Compression. Multimedia file formats Image: JPEG, PNG, GIF, TIFF, PNG Audio: MP3, WAV, MP4 etc Video: MP4, AVI, WMV, FLV, MPEG, MOV etc Image editing techniques	1. Explore interfaces of editing tool such as adobe photoshop a. Customizing Workspaces. b. File Handling c. Setting size and resolution d. parameters. e. Importing files. f. Working with Layers g. Explore tool box h. Explore Filters 2. Perform photo compositing 3. Create Personal Visiting card
12	3	4,5	Design a graphic • Cards • Flyer • Banner • Advertisement Using blend modes create • Logo • Poster Creating custom shape & text Wrapping	1. Create Invitation for College fest 2. Create Banner for College admission promotion activities 3. Create social media graphic for college social media and placement accounts.
13	5,1,3	4,5,7	Artificial Intelligence Definition of AI Difference between AI and Human Intelligence Types of AI • Narrow AI (Weak AI) • General AI (Strong AI) • Superintelligent AI Real-World Applications and Use Cases of AI: AI in Industry • Healthcare • Finance • Automotive • Retail AI in Consumer Products	Know the usage of free AI tools: Learning and Study Tools: ChatGPT (Free Plan): (https://chat.openai.com/) Google Gemini https://gemini.google.com/ Wolfram Alpha (https://www.wolframalpha.com/) Writing and Language Tools: Grammarly (Free Plan) (https://www.grammarly.com/) Quillbot(https://quillbot.com/) DeepL Translator(https://www.deepl.com/) LanguageTool(https://languagetool.org/) Creativity and Visual Tools: Canva (Free for Education)

			Smart Assistants: Siri, Alexa, and Google Assistant. AI in Social Media: Content recommendation, ad targeting Entertainment: Netflix and Spotify etc. Generative AI	Runway ML (Free Plan) Problem Solving Copilot: Assists with coding and programming tasks, offering real-time suggestions and error corrections to enhance learning How AI works: Google Teachable Machine https://teachablemachine.withgoogle.com/
--	--	--	---	---

4. Learning Resources

Week	Resources
1	1. https://edu.gcfglobal.org/en/basic-computer-skills/ 2. https://www.geeksforgeeks.org/computer-fundamentals-tutorial/ 3. Any youtube demo videos 4. ChatGPT 5. NCERT School books
2	1. https://edu.gcfglobal.org/en/topics/internet/ 2. https://www.geeksforgeeks.org/basics-computer-networking/ 3. Any youtube demo videos 4. ChatGPT 5. School books
3	1. https://edu.gcfglobal.org/en/topics/onlinesafety/ 2. https://www.geeksforgeeks.org/cyber-security-types-and-importance 3. https://www.w3schools.com/cybersecurity 4. Introduction to Cybersecurity: Netcad.com course 5. ChatGPT 6. School books
4	1. https://edu.gcfglobal.org/en/topics/onlinesafety/ 2. https://www.geeksforgeeks.org/cyber-security-types-and-importance 3. https://www.w3schools.com/cybersecurity 4. https://www.cisco.com/c/en_in/products/security/common-cyberattacks.html#~types-of-cyber-attacks 5. Introduction to Cybersecurity: Netcad.com course 6. ChatGPT 7. School books
5	1. NCERT class 11 Computer Science 2. Youtube videos
6	1. NCERT class 11 Computer Science 2. Youtube videos
7,8	1. https://scratch.mit.edu/explore/projects/tutorials/ 2. https://developers.google.com/blockly
09	1. https://www.geeksforgeeks.org/cloud-computing-tutorial/ 2. https://cloud.google.com/ 3. https://aws.amazon.com/education/awseducate/ 4. https://edu.gcfglobal.org/en/topics/usingthecloud/
10	1. https://www.javatpoint.com/iot-internet-of-things 2. https://www.ibm.com/topics/internet-of-things

	3. https://www.tinkercad.com/ 4. https://mblock.cc
11,12	1. https://helpx.adobe.com/in/photoshop/user-guide.html 2. https://webneel.com/ 3. https://clippingpathindia.com/ 4. https://www.photoshopessentials.com/basics/ 5. https://www.befunky.com/
13	1. NCERT Informatics Practices for Class 11 2. https://www.geeksforgeeks.org/What-is-ai-artificial-intelligence/ 3. https://teachablemachine.withgoogle.com/ 4. https://scratch.mit.edu/search/projects?q=AI 5. https://gemini.google.com/ 6. https://www.perplexity.ai/ 7. https://chatgpt.com/ 8. https://www.meta.ai/ 9. https://copilot.microsoft.com/ 10. https://www.canva.com/ai-image-generator/ 11. https://www.slido.com/

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. Theory Test model question paper

Program	Information Science and Engineering			Semester -1	
Course Name	IT skills			Test	III
Course Code	25CS01I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	Questions		Cognitive Level	Course Outcome	Marks
Section - 1					
1	a) Write Algorithm to accept numbers till the user enters 0 and then find their average. 15 Marks		A	2	25
	b) Describe Steps for Problem Solving with example. 10 Marks		A	2	
2	1. a) Write Flowchart to draw following pattern on screen * ***** ***** ***** * 15 marks		A	2	25
	b) List different symbols used in flowchart, list and describe their functions 10 marks		A	2	
Section – 2					
3	a) Explain any 2 loop and decision blocks with example 5 Marks		U	2	25
	b) What is IoT? Illustrate with example. 10 Marks		U	5	
	c) What are cloud service models? Illustrate its applications with example. 10 Marks		A	5	
4	a) What is block coding? Illustrate its advantages 5 marks		U	2	25
	b) What are cloud deployment models? Illustrate its applications with example 10 marks		U	5	
	c) List and explain applications of IoT with example 10 Marks		U	5	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.					

Signature of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	1
Course Name	IT skills			Test	II
Course Code	25CS01I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
1. Design Algorithm, flowchart and implement program to check largest of 3 numbers in MIT scratch				2	50
Scheme of assessment					
a) Algorithm					10
b) Flow Chart					10
c) Scratch Application block code design and implementation					20
d) Demonstration					10
TotalMarks					50

Sign of the Course Coordinator

Signature of the HOD

9. SEE- Model Practice Question Paper

Program	Information Science and Engineering		Semester	1
Course Name	IT skills	Course Code : 25CS01I	Duration	180 min
Questions			CO	Marks
1. i. Design Algorithm, flowchart and implement program to a simple apple catching game in MIT scratch ii. save your algorithm, flowchart in cloud iii. Create a digital poster for your game and demonstrate			1,2,3,5	50
Scheme of assessment				10
a) Algorithm and flowchart				10
b) Scratch Application block code design and implementation				10
c) Use of cloud services				10
d) Multimedia design				10
d) Demonstration				10
Total Marks				50

Signature of the Examiner

2) Signature of the Examiner

10. Rubrics for Assessment of Activity(Qualitative Assessment)

Sl. No.	Dimension	Excellent (10)	Good (8)	Satisfactory (6)	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
TOTAL						40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

11. Model Questions

Week	T/P	Questions
1	T	- What is Computer? Explain digital computer with block diagram - Illustrate generations and types of computer - List and describe functional blocks of computer - What are different types of memory in digital computer? Illustrate each with example - Differential between main memory and secondary memory - Differential between Input and Output devices - List and explain different digital computer cables and connectors - What is an operating system? Illustrate uses of operating system - Describe different types of software with examples - List the factors to be considered while purchasing a computer
	P	Connect parts of computer, create a folder for yourself & install scratch software in it.
2	T	- What is computer network? Illustrate types - Differentiate LAN, MAN, WAN - Differentiate Physical and Logical address - Illustrate key network devices - What is Internet? List applications - Illustrate Client – server model - Explain browser and web server - What is domain name system? Illustrate domain names with example - What is an URL? Illustrate with example - Illustrate search engine with example

	P	1. Create E- Mail account and demonstrate the services it offer 2. Download and install any web browser 3. Design and host a personal website for yourself 4. Configure IP address for your computer
3	T	1. What is Cyber security? Illustrate Confidentiality, Integrity, Availability. 2. What is data? Describe types of data 3. List measures to protect your personal and organizational data 4. Where is your social media data is stored? Did you give permission to store and share your data? Discuss 5. Discuss important settings in your home wifi settings. 6. What are the risks while using public Wifi? 7. Discuss Basic security issues in your mobile phones and personal computers 8. Discuss importance of two factor authentication
	P	6. Demonstrate Two-factor authentication 7. Demonstrate Browser settings 8. Demonstrate smart phone App permissions and settings. 9. Install Firewall and demonstrate settings. 10. Demonstrate Wireless network security settings
4	T	1. What is virus ? illustrate with example. 2. Differential between ransomware and spyware 3. List and explain different types of malwares 4. What is Phishing? Illustrate with example 5. Discuss social engineering 6. Illustrate denial of service attack. 7. Illustrate On-path attack, Man in the middle attack, Eavesdropping, Spamming, Identity theft, Password attacks, SEO Poisoning. 8. What is an antivirus software? Discuss its importance 9. Differential between HTTP and HTTPS 10. What is firewall? How does it protect your device? 11. Illustrate cookies, explain role of cookies in security risk. 12. What is Cyberbullying?
	P	4. Install of Antivirus software and demonstrate settings 5. Demonstrate the process of updating Browser Plug-ins and settings 6. Demonstrate Privacy settings in social media accounts. 7. Perform Security audit (Antivirus, strong password, firewall, app settings and permissions, unwanted apps) of all your smart phone.
5	T	1. Describe Steps for Problem Solving with example. 2. Illustrate Algorithm with an example. 3. Write Algorithm to accept numbers till the user enters 0 and then find their average 4. Write Algorithm to draw following pattern on screen <pre> * ***** ***** ***** *</pre> 5. Write Algorithm to Calculate and print the sum of 'N' numbers 6. Write Algorithm to check whether a number is odd or even

	P	1. Write Algorithm to accept numbers till the user enters 0 and then find their average. 2. Write Algorithm to draw following pattern on screen <pre> * ***** ***** ***** *</pre> 3. Write Algorithm to Calculate and print the sum of 'N' numbers. 4. Write Algorithm to check whether a number is odd or even.
6	T	2. List different symbols used in flowchart, list and describe their functions. 3. Write Flowchart to accept numbers till the user enters 0 and then find their average find their average. 4. Write Flowchart to draw following pattern on screen <pre> * ***** ***** ***** *</pre> 5. Write Flowchart to Calculate and print the sum of 'N' numbers. 6. Write Flowchart to check whether a number is odd or even.
	P	1. Write Flowchart to accept numbers till the user enters 0 and then find their average find their average. 2. Write Flowchart to draw following pattern on screen <pre> * ***** ***** ***** *</pre> 3. Write Flowchart to Calculate and print the sum of 'N' numbers. - Write Flowchart to check whether a number is odd or even.
7,8	T	1. What is block coding? Illustrate its advantages 2. Explain any 2 loop and decision blocks with example 3. Design and implement program to check largest of 3 numbers. 4. Design and implement program calculate sum of n even numbers.
	P	1. Design and implement a scratch project to move a object 10 steps and say hello. 2. Design and implement program to check largest of 3 numbers. 3. Design and implement program calculate sum of n even numbers. 4. Design a simple game using the MIT scratch tool.
9	T	1. What is cloud computing? Illustrate its applications with example 2. What are cloud deployment models? Illustrate its applications with example 3. What are cloud service models? Illustrate its applications with example 4. Differentiate between SaaS / PaaS and IaaS 5. Differentiate between Public / Private / Hybrid cloud models
	P	1. Demonstrate creating folder, organizing files and editing documents in google drive. 2. Collect attendance using google form, edit the data in google drive and

		demonstrate its advantages
10	T	1. What is IoT? Illustrate with example. 2. Illustrate components of IoT. 3. Discuss features of IoT. 4. List and explain applications of IoT with example.
	P	1. Create a simple visual block code to blink LED in Arduino board using visual block code, upload code to Arduino board and demonstrate. 2. Create a Traffic signal controller with 3 LED (RED, YELLOW and GREEN), upload code to Arduino board and demonstrate.
11	T	Illustrate Building blocks of multimedia. Write a note on image resolution and how is this connected to image file size Discuss major multimedia file formats.
	P	2. Explore interfaces of editing tool such as adobe photoshop i. Customizing Workspaces. j. File Handling k. Setting size and resolution l. parameters. - m.Importing files. n. Working with Layers o. Explore tool box p. Explore Filters 2. Perform photo compositing 3. Create Personal Visiting card
12	T	1. Illustrate steps in creating multimedia banner. 2. Illustrate steps in creating Logo or poster.
	P	1. Create Invitation for College fest 2. Create Banner for College admission promotion activities 3. Create social media graphic for college social media and placement accounts.
13	T	1. What is Artificial Intelligence? Describe a real world example. 2. Illustrate AI types and Use. 3. Illustrate applications of AI. 4. Write note on generative AI.
	P	1. Google Teachable Machine : Demonstrate classification of male and female 2. Gemini: Demonstrate its use for generating answers, creating summary etc. 3. ChatGPT: Demonstrate its use for creating project report. 4. Copilot: Demonstrate using copilot to design personal website 5. Canva AI: Create banner for your college fest/admission promotion. 6. Grammarly : Demonstrate Correcting your activity report for any grammatical errors

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	Intel i5, 4 GB RAM, 1 TB SDD	30
02	Internet Connectivity		
03	Arduino board	Basic	10
04	LED		20



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	1
Course Name	Basics of Digital Logic and Computer Organization	Type of Course	Integrated
Course Code	25CS11I	Contact Hours	8 Hours per week
Teaching Scheme	4:0:4	Credits	6
CIE Marks	50	SEE Marks	50 (Theory)

1. Rationale

This course is designed to provide a comprehensive foundation for understanding how computers operate at their core. By studying fundamental digital circuits, students acquire a clear understanding of how data is represented, manipulated, and stored, alongside the mechanisms through which logical operations are performed to solve computational problems. Furthermore, key concepts in computer organization empower students to explore the architecture, functionality, and collaboration of essential hardware components that enable the seamless execution of instructions.

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

CO-01	Analyze and comprehend the data representation in digital systems.
CO-02	Design a digital circuit using suitable gates for the given scenario
CO - 03	Simulate and examine the functionalities of combinational and sequential circuits using appropriate software tools.
CO - 04	Explain the basic architecture and functioning of a computer
CO-05	Explain the interaction of hardware components in execution of an instruction.

3. Course Content

Week	CO	PO	Lecture(4HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1	Data representation Numerical – Binary, octal and hexadecimal BCD, ASCII, Unicode Conversions between number systems.	Organize and play games like: Conversion Bingo game Number Conversion Relay Base Conversion Escape Room Binary to ASCII Decoder Game Binary Jigsaw Puzzle <u>Binary Game (cisco.com)</u> <u>Binary Bonanza! Binary Number game - Fun, Free, Online Way to Learn Binary (penjee.com)</u>
2	1	1,2	Negative Numbers and Binary arithmetic Signed magnitude, 1's complement, and 2's complement representation. Floating point representation Arithmetic operations in binary (addition, subtraction)	<u>Learn Binary Arithmetic - Binary Tutorial (ryanstutorials.net)</u> Treasure hunt game
3	2	1,2,3	Logic gates Overview of digital logic design and its applications in computer science and engineering. Logic gates-Definition, symbol and truth table and their applications– NOT, OR, AND, NOR, NAND, XOR, XNOR	<u>Every Circuit or similar simulator can be used</u> Explore a simulator interface Examine the functionality of each gate using simulator Design simple circuits using logic gates. Visualize gate operations using LED-based kits or online simulators. https://www.youtube.com/watch?v=HjneAhCy2N4
4	2	1,2,3	Boolean Algebra and Simplification Laws, Rules and theorem Boolean expressions – SOP, POS	<u>EveryCircuit or similar simulators</u> Test and analyze the Boolean laws and rules
5	2	1,2,3	Circuit design using Boolean expression Derive Boolean expression from truth table	Utilize simulators to create and test digital circuits based on Boolean expressions. Analyze how the simplifying expression impacts circuit performance and complexity.
6	3	1,2	Combinational Logic Circuits Arithmetic Circuits: half adder, full adder Half-Subtractor and Full-Subtractor. Data Processing Circuits: Multiplexer, Demultiplexer, Encoder and Decoder Comparator Circuits: Single-bit and multi-bit comparators.	<u>Virtual Labs (vlabs.ac.in)</u> <u>simulator.io Anonymous board</u> Examine the working of Half-Adder and Full-Adder circuits. Multiplexers and Demultiplexers. Comparator
7	3	1,2	Sequential Logic Circuits Flip-Flops – definition, types, applications	simulator.io Anonymous board <u>Examine the functionality of different flip flops</u>

			Registers- definition and shift register types Counters: Asynchronous (Ripple) and Synchronous Counters. Compare combinational and sequential circuits	<u>Simulation of</u> Flip-Flop-based counters and shift registers. Simulate flip-flops and counters using software like Tinkercad or Logisim.
8	4	1,2	Computer Organization Definition- computer architecture and computer organization. Von Neumann Architecture Central Processing Unit (CPU) – components and performance metrics.	<u>The central processing unit (CPU): Its components and functionality Enable Sysadmin (redhat.com)</u> CPU-OS simulator Get familiar with environment of CPU with simulator https://www.youtube.com/watch?v=d86ws7mQYIg https://www.youtube.com/watch?v=h9Z4oGN89MU
9	4	1	Instruction Set Architecture (ISA) – definition and role Instruction types Addressing modes Interrupts - types, handling mechanism	Demonstrate how the CPU handles interrupts (both hardware and software) and returns to normal execution. https://www.youtube.com/watch?v=PVNAPWUxZ0g
10	4	1	Specialized Processors Emergence of specialized processors for specific tasks (GPU, TPU, NPU). Components of a GPU - Compute Units, Cores and Functionalities of GPU	https://www.youtube.com/watch?v=h9Z4oGN89MU Compare CPU, GPU, TPU and NPU
11	5	1	Memory and its Hierarchy Units of memory Types - Primary Memory, Secondary Memory and Tertiary Memory Memory Hierarchy Applications of Memory Types	https://www.youtube.com/watch?v=h9Z4oGN89MU https://www.youtube.com/watch?v=5Mh3o886qpg
12	5	1	Input/output (I/O) Input devices Output devices I/O Ports – Definition, common ports , standards and protocols	Locate and identify common I/O ports on a computer Identify I/O devices connected to a computer
13	5	1	System Buses and Communication System bus and types of buses Bus Architecture: - parallel and serial Bus protocols and standards: PCI, USB, SATA.	Identify the system buses in the computer

4. Learning Resources:

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

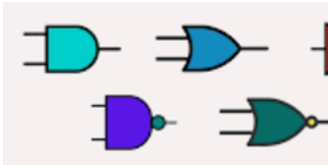
6. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Theory	90	50	20

7. CIE Theory Test model question paper

Q12 Theory Test Model Question paper

Program	Information Science and Engineering			Semester - I		
Course Name	Basics of Digital Logic and Computer Organization			Test	I/ III	
Course Code	25CS11I	Duration	90 min	Marks	50	
Name of the Course Coordinator:						
Note: Answer any one full question from each section. Each full question carries equal marks.						
Q.No	Questions			CL	CO	MARKS
Section - 1						
1	a) Group the following numbers into their respective number systems (Decimal, Binary, Hexadecimal). If a number can belong to multiple systems, treat it as decimal by default. Explain your reasoning for each classification: 62, 1FA,0101011,75 – (10M) b) Represent the decimal number -34 in binary using the following methods (sign-magnitude, 1's complement, and 2's complement) for an 8-bit representation. – (6M) c) Perform binary addition on the following two 8-bit numbers:10110011 and 11001101 –(4M) d) A digital clock uses Binary-Coded Decimal (BCD) to display numbers on a seven-segment display. Explain why BCD is used instead of pure binary and show how the number 25 is represented in BCD format. –(5M)			2	1	25
2	a) Convert the decimal number 87 into its equivalents in binary, octal, and hexadecimal number systems. Explain the steps you took for each conversion. – (10M) b) A temperature sensor gives readings in an 8-bit two's			2	1	

	complement format, where positive numbers represent temperatures above zero and negative numbers represent temperatures below zero. If the sensor reads 11100100, convert this reading to decimal to find the temperature. (6M) c) Convert each character of the word “HELLO” to its ASCII binary representation. (5M) d) Perform the subtraction $100101 - 1011$ using two’s complement binary arithmetic.(4M)			
Section – 2				
3	a) Which gates would you use to design a circuit for these scenario, justify your selection (12M) - The output should be 1 only when at least one input is 1. - The output is 1 only when the two inputs are the same. - In a control system, you need an output of 1 only when all inputs are 0. - For a security system, if you want the output to be 1 only when both switches are in the off position. - Output is 1 if and only if the inputs are different. - Output is 0 when both inputs are 1. b) Construct the truth table for a 3-input AND gate, where the inputs are A, B, and C, and the output X is defined as: (5M) $X = A \cdot B \cdot C$ c) You are part of a team solving a digital treasure hunt. To unlock the treasure chest, you need to figure out the secret combination using a circuit made entirely of NAND gates. Follow the clues, determine the outputs, and uncover the secret combination. Clue 1 : Single NAND Gate -You find a locked door controlled by a single NAND gate. The inputs A and B are both connected to switches. The door opens only if the output X=1. What are the possible combinations of A and B to open the door? (8M)			25
4	Identify the following gates and construct truth table for each (12M)  b) A circuit consists of two inputs, A and B. The output is high only when both A and B are either high or low. Which logic gate is used in this circuit, and why? –(5M) c) You’ve been given a mysterious circuit that encodes or decodes a secret binary message using XOR gates. Your task is to figure out the input, the circuit logic, and the hidden message. If The binary message $M = 11001$ and the secret key $K = 10101$ what is the encoded message X To decode the message, the encoded message X is passed through the same XOR gate with the original key $K = 10101$. what is the decoded message M. - (8M)			

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Sign of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	1
Course Name	Basics of Digital Logic and Computer Organization			Test	I/ III
Course Code	25CS11I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
A security system outputs F if: Door 1 is locked (A) OR Door 2 is locked (B), AND Alarm is OFF (C). Write the Boolean expression, simplify it, construct truth table and verify it using simulator. Design a 2-bit magnitude comparator that compares two 2-bit binary numbers, A and B, and generates the following outputs: A>B: Output is 1 if A is greater than B. A=B: Output is 1 if A is equal to B. A<B: Output is 1 if A is less than B. Tasks: Construct the truth table for the 2-bit magnitude comparator, considering all possible values of A and B. Derive the logic expressions for each output (A>B, A=B, A<B). Implement the circuit using basic logic gates (AND, OR, NOT, XOR, etc.). Validate the circuit by simulating it with a tool.				2,3	50
Scheme of assessment a) Understanding and Problem Analysis - 10 b) Truth Table and Logic Expressions - 15 c) Circuit Design and Implementation - 15 d) Presentation and Documentation – 10					
Total Marks					50

Signature of the Course Coordinator

Signature of the HOD

9. SEE–Model Theory Question Paper

Program		Information Science and Engineering	Sem	1	
Course Name		Basics of Digital and computer organization	Marks	50	
Course Code		25CS11I	Duration	90 Min	
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q N o	Questions		Cognitive Levels	Course Outcomes	Marks

Section -1				
1	Perform the following conversions: Convert $10111010_{(2)}$ to its hexadecimal equivalent. Convert $7F3_{(16)}$ to its decimal equivalent. Convert $89_{(10)}$ to its BCD representation. Explain the concept of character data representation with examples. Perform the subtraction $100101_{(2)} - 1011_{(2)}$ using two's complement binary arithmetic. Show all intermediate steps.	2	1	10
2	a) Explain the characteristics of the following number systems with examples: Binary (Base-2) , Decimal (Base-10) Octal (Base-8), Hexadecimal (Base-16) b) Add the following two 8-bit binary numbers and provide the final result: $10110011_{(2)}$ $11001101_{(2)}$ c) Convert the decimal number -47 into an 8-bit two's complement binary representation. Show all steps involved in the conversion.	2	1	10
Section -2				
3	a) A home security system has three sensors: one on the door (D), one on the window (W), and one on the motion detector (M). The alarm (A) should go off if: Either the door or the window sensor is triggered, OR The motion detector is triggered while the door is closed. Write the Boolean expression for this alarm system, simplify it if possible, and draw the logic circuit.	3	2	10
4	A fan is controlled by temperature sensors (T) and humidity sensors (H). The fan turns on if: The temperature is high (T), OR Both the temperature and humidity are at moderate levels. Formulate the Boolean expression to represent the fan control system. Simplify it if possible and draw the circuit using logic gates.	3	2	10
Section -3				
5	Classify different types of combinational circuits and provide examples for each category. Construct truth tables for a half-adder circuit, showing all possible input combinations and their corresponding outputs (sum and carry).	2	3	10
6	Construct truth tables for D and JK flip-flops, clearly indicating the input-output relationships. Interpret the behavior of these flip-flops based on their truth tables. Compare combinational circuits and sequential circuits, highlighting their key differences in functionality, design, and applications.	2	3	10
Section -4				

7	a) Explain the operational mechanics of the Von Neumann architecture, including the fetch-decode-execute cycle . b) Explain the concept of addressing modes and categorize the common types with examples.	2	4	10
8	a) Categorize the common types of addressing modes and provide examples for each. b) Describe the key components of a Graphics Processing Unit (GPU).	2	4	10
Section -5				
9	a) Explain the functions of each bus type in the data transfer process within a computer system. b) Classify the various types of input and output devices used in computing systems c) A user needs to store a video library of 500 HD movies, where each movie is 5 GB. How much storage space is required in terms of terabytes?	2	5	10
10	a) Provide examples for each type of memory and describe their specific functions and applications within the system. b) Explain the role of the system bus in computer communication, including how it connects different components	2	5	10

11. Rubrics for Assessment of Activity (Qualitative Assessment)

Sl. No.	Dimension	Excellent	Good	Satisfactory	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
TOTAL						40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers		15
02	Internet Connection	100mpbs	

SEMESTER 2



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	2
Course Name	Thinking Programming with Python	Type of Course	Integrated
Course Code	25CS21I	Contact Hours	8 Hours per week
Teaching Scheme	4:0:4	Credits	6
CIE Marks	50	SEE Marks	50 (Theory)

1. Rationale:

The course aims to combine foundational problem-solving skills with practical programming experience in Python. This integrated approach enables learners to develop a logical mindset while gaining hands-on coding experience in one of the world's most versatile and beginner-friendly programming languages. The course goes beyond teaching programming syntax. It's an essential step toward developing logical, computational, and technical proficiency for success in the digital age.

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

CO-01	Apply computational thinking to solve the given problem and illustrate the solution as an algorithm
CO-02	Develop a programmer's mindset through the application of fundamental programming concepts.
CO-03	Code and execute a program to solve the given problem using python concepts
CO-04	Identify and resolve both syntactical and semantic errors.
CO-05	Apply modular programming approach to optimize the program.

3. Course Content

Week	CO	PO	Lecture(4HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1,2	Computational Thinking Introduction, Components, Importance and Applications of CT in various fields. Decomposition: Consider a problem statement to learn decomposition (refer table 1 for examples) – Pattern recognition: Analogy; Classification; Sequencing; Ranking; Series	Organize Games and Activities to instill Computational Thinking
2	1	1,2	Algorithmic thinking Problem Types for Algorithmic Solutions [definition and examples] ▪ Searching Problems ▪ Sorting Problems ▪ Optimization Problems ▪ Graph Problems ▪ String Processing Problems ▪ Numerical Problems ▪ Combinatorial Problems ▪ Cryptographic Problems Algorithm Representation: Natural Language, Pseudocode, Flowchart <i>[Note: Sorting Algorithms: designed to arrange data in a specific order Examples: Bubble Sort, Merge Sort, Quick Sort]</i>	For a given problem ▪ Identify the key components (input, output and logic) of an algorithm ▪ Develop a step-by-step algorithm to solve it.
3	2	1	Introduction to Programming What is programming? ; Why we need Programming; How to Think like a programmer; Programming Paradigms; Programming Languages and their Paradigms	Organize Games and Activities to instill Programmer Thinking
4	2	1	Basic Programming Concepts: Syntax ; Tokens and types; Variable -Rules for creating variables; constants; datatypes; errors ; comments ; Best programming practices;	Setup and get familiar with Your Development Environment such as VsCode
5	2	1	Introduction to python programming: Features and applications ; Data types (primitive); Assignment statement; Type conversion;	Practice Programs to understand datatype, and their conversion.
6	3,4	1	Input and output statements: input (); print (); Formatting output - string concatenation, format() and f-strings	Practice Programs to understand reading input and formatting output
7	3,4	1	Operators and their Precedence Expressions	Practice Programs to understand Operators and Expressions

8	3,4	2,3	Flow control Conditional statements: if, if-else, if-elif, match case	Practice programs to understand the concept of conditional statements
9	3,4	2,3	Iterative statement – for loop: structure of the for loop using range();concept of break and continue with the for loop;	Practice programs to understand the concept of for loop
10	3,4	2,3	Iterative statement - While loop : structure; concept of Break, continue, pass statements with while loop;	Practice programs to understand the concept of while loop
11	3,4	2,3	Nested loops : Use cases of nested loops; control flow in nested loops using break, continue, and else;	Practice programs to understand the concept of nested loops
12	3,4,5	2,3	Functions : Need for functions ; create function ; function call: with and without arguments; return statements; Variable scope	Practice programs to understand the concept of functions
13	3,4,5	2,3	Modules & Packages : Significance of modules and packages; Import and use built-in modules; Create new module; Create a package with multiple modules; import statement (import, import...as .., from....import, import *)	Practice programs to understand the concept of modules and packages

4. Learning Resources:

week	Suggested program/ activity list
1	Case based learning: - Improving the Transport System in the Countryside - Analyzing the Reach of Government Schemes - Smart Irrigation System for Optimal Water Usage - Impact of Social Media on Students
2	Problem / use case based learning Devise an Algorithm and draw flowchart for problems such as - Swapping two values - Finding largest / smallest among two/three number - Computing area/ perimeter of given shape (circle, triangle, rectangle, square) - Metric conversion (meter – KM, pound – kilo gram, Celsius – Fahrenheit) - Determine given number is even or odd, positive or negative.
3	<u>Think Like a Programmer - Google Books</u> <u>Think Like a Programmer: Introduction (youtube.com)</u> General problem solving techniques- refer the book or youtube videos <u>https://www.codecademy.com/resources/blog/how-to-think-like-a-programmer/</u>
4	Prepare Your Development Environment: - Download and install the necessary compiler or interpreter - Python: Download from python.org. - Java: Install the Java Development Kit (JDK) from Oracle or OpenJDK. - Verify the installation by checking the version using the terminal/command prompt - Install an IDE (for python VSCode or Pycharm) - Open your IDE and explore its features:

	▪ Create a new project. ▪ Write a simple "Hello, World!" program. ▪ Learn to run and debug your code. ▪ Explore useful features like syntax highlighting, auto-completion, and integrated terminal.
5	Create variables, assign values, and display their data types: a. Create variables of different data types and assign values b. Display the values and their respective data types using the type() function. Assignment Statements: a. Single variable assignment: Assign a value to a single variable. b. Multiple variable assignment: Assign values to multiple variables in one statement. c. Assign the same value to multiple variables. Type Conversion a. Demonstrate Python's ability to convert types automatically during operations. b. Use functions like int(), float(), str(), or bool() to explicitly convert data types.
6	1. Write a program that: a. Accepts user input for name and age. b. Prints a formatted message using different string formatting methods. 2. Read message from the user and Format it with different methods and display. 3. Demonstrate swapping the values(numerical) of two variables using a temporary variable.
7	1. Write python equivalent expressions for math expressions such as a. $f = ax + b$ b. $f = a^2 + b^2 + 2ab$. c. $f = a^3 + b^3 + 3ab(a + b)$ d. $area = \pi r^2$ e. $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ f. $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 2. Translate textual problem statements into Python expressions 3. Write a program that: a. Accepts two numbers from the user. b. Converts them to integers (if they are not already). c. Performs arithmetic operations (add, subtract, multiply, divide) and displays the results with the data types and appropriate message. 4. Calculate the total cost of an item after applying a 15% discount to its original price of Rs150. 5. Calculate the compound interest on a principal of Rs10000 at an annual interest rate of 5% for 3 years, compounded annually. 6. Compute area of triangle, rectangle. 7. Swapping values of two variables without using a temporary variable
8	1. Create a program to manage airline ticket bookings based on the passenger's selected class: "economy", "business", or "first". a. Prompt the user to input details such as name, age, destination, and class preference (economy, business, or first). b. Validate inputs (e.g., ensure the name contains only letters, age is a positive integer, and

	class selection is valid). c. Display appropriate error messages for invalid entries and allow the user to re-enter the data. 2. Build a system to suggest clothing based on the temperature. a. If temperature $\geq 30^{\circ}\text{C}$, suggest "Wear light clothes." b. If $20^{\circ}\text{C} \leq \text{temperature} < 30^{\circ}\text{C}$, suggest "Wear moderate clothing." c. If temperature $< 20^{\circ}\text{C}$, suggest "Wear warm clothes." 3. Implement a discount system for an e-commerce website. a. If the total purchase is $\geq \text{Rs}5000$, apply a 20% discount. b. If $\text{Rs}2000 \leq \text{total} < \text{Rs}5000$, apply a 10% discount. c. If total $< \text{Rs}2000$, no discount is applied. 4. Create a program that simulates an ATM machine to check: a. If the entered PIN is correct, allow the user to proceed. b. Validate that the withdrawal amount doesn't exceed the account balance. c. Ensure that the withdrawal amount is in multiples of a specific denomination (e.g., Rs100).
9	1. Generate numbers from 1 to 10 using range(). 2. Generate numbers for a given range. 3. Write a program to display square numbers from 1 to 10. 4. Write a Python program that accepts an integer input from the user and determines whether the number is a prime number or not. If the number is prime, display an appropriate message; otherwise, indicate that it is not prime. 5. Write a Python program that accepts a message and a number from the user. The program should then print the specified message the given number of times. 6. Write a Python program to calculate the factorial of a given number. The program should: a. Accept a non-negative integer as input. b. Use an iterative approach to compute the factorial. c. Handle invalid inputs (e.g., negative numbers or non-numeric inputs) by displaying appropriate error messages. 7. Evaluation of mathematical series like $S=1+2+3+\dots+n$ $S=1^2+2^2+3^2+\dots+n^2$ $H_n=1+1/2+1/3+\dots+1/n$
10	1. Implement the Euclidean algorithm to find the GCD of two integers using a while loop. 2. Collect daily weather data (e.g., temperature, humidity) from the user. Allow data entry to continue until the user decides to stop or inputs a sentinel value. 3. Simulate a loan repayment system where a borrower pays a fixed amount each month. Continue reducing the loan balance until it is fully paid off, and display the balance after each payment. 4. Write a program that simulates a countdown timer. Start with a given number of seconds and decrement until it reaches zero, displaying the countdown at each step. 5. Write a program that takes an integer and calculates the sum of its digits using a while loop. 6. Write a program that simulates a simple user authentication system. The program should: a. Repeatedly ask the user to input their username and password. b. Verify the entered credentials against a pre-defined username and password. c. Allow the user up to three attempts to enter the correct credentials.

	d. Provide appropriate feedback for each attempt (e.g., "Incorrect username or password"). e. If the correct credentials are provided within three attempts, display a success message and terminate the program. f. If all three attempts are exhausted, display a failure message and terminate the program.
11	1. Write a program to identify all prime numbers within a user-defined range. The program should: a) List all the prime numbers within the range. b) Calculate and display the total count of prime numbers found. c) Compute and display the sum of these prime numbers. 2. Write a Python program to perform the following tasks for a user-defined range of integers: a) Identify and list all non-prime numbers within the given range. b) Categorize the non-prime numbers into even and odd. c) Count the total number of even and odd non-prime numbers. d) Calculate and display the sum of even and odd non-prime numbers separately. 3. Write a Python program to perform the following tasks for a user-defined range of integers: a) Identify and list all palindrome numbers within the specified range. b) Count the total number of palindrome numbers found. c) Display the results in a user-friendly format. 4. Write a program to generate and display a multiplication table for numbers 1 to 10. 5. Generate star or number patterns like a pyramid or diamond shape.
12	1. Write simple functions for arithmetic calculations (e.g., addition, factorial). 2. Define a function <code>assign_priority()</code> for ticketing system which assigns priorities to support tickets based on the issue type (low, medium, high) and returns a priority level. 3. Define function <code>calculate_interest()</code> for banking system, to calculate the interest on savings accounts based on the principal amount, rate, and time period. 4. An online store offers a discount based on the total purchase amount. If the customer's total purchase exceeds a certain threshold, they get a percentage discount. Define a function <code>apply_discount()</code> that accepts the total amount and returns the final price after discount.
13	1. Create a module that includes a function to calculate the area of a circle and use it in another script. 2. Create a package for a small project, such as a calculator application with modules for arithmetic, trigonometric, and logarithmic operations.

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
	CIE-5 Portfolio evaluation of				

5	all the activities through Rubrics	1-13		50	
				Total	50 Marks

6. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Theory	90	50	20

7. CIE Theory Test model question paper

Program	Information Science and Engineering			Semester – 2	
Course Name	Thinking Programming with Python			Test	I/III
Course Code	25CS21I	Duration	90 min	Marks	50

Name of the Course Coordinator:

Note: Answer any one full question from each section. Each full question carries equal marks.

Q.No	Questions	Cognitive Level	Course Outcome	Marks
Section – 1				
1	a. You are tasked with organizing a class party. The goal is to ensure everything is well-prepared, including food, decorations, activities, and invitations. Use decomposition to break down this problem into smaller, manageable tasks. – (10 M) b. Identify the pattern and provide an explanation to complete the series. (10M) 1. 3, 6, 11, 18, 27, _____, _____, _____, _____ 2. 1, 1, 2, 3, 5, 8, _____, _____, _____, _____ 3. 2, 4, 12, 48, _____, _____, _____, _____ 4. A2, B4, C8, D16, _____, _____, _____, _____ c. Explain the four pillars of computational thinking. – (5M)	2,3	1	25
2	a. You are tasked with organizing a birthday party for your best friend. The goal is to ensure the party runs smoothly. To accomplish this, break the task into smaller, manageable steps and write down the plan. – (10M) b. A magical machine only accepts even numbers and rejects odd ones. If you input an even number, the machine lights up green; if you input an odd number, it lights up red. Write an algorithm and draw flowchart to help a wizard decide whether a number can be accepted by the machine or not.- (7M) c. Identify different types of problems and provide an example	2,3	1	

	algorithm to solve each. - (8M)			
Section – 2				
3	a. Explain the concept of a variable. Why are variables essential in programming? b. Define tokens. List and explain the different types of tokens with examples. c. List the rules for naming variables in Python. Provide examples of valid and invalid variable names. d. Why are constants important in programming? Illustrate with an example where a constant would be beneficial. e. What are syntax errors, logical errors, and runtime errors?	2	2	25
4	1. How can learning programming enhance one's logical thinking, creativity, and problem-solving abilities? 2. Explain the concept of "thinking like a programmer" and its importance in solving real-world problems. 3. Explain how identifying patterns in problems can lead to efficient solutions. Use an example to demonstrate this process. 4. Trace the evolution of programming paradigms through the generations of programming languages. 5. A team is building a web application with a real-time chat feature. Discuss which programming paradigms and languages might be best suited for this task and why.	2	2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	2
Course Name	Thinking Programming with Python			Test	II/IV
Course Code	25CS21I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
a. Write a program that performs the following tasks: a. Generate Random Numbers: Generate a random number within a given range [a, b], where a and b are user inputs. b. Determine non-prime Number: Determine that generated random number is not prime. c. Classify non-prime Number: Further classify numbers into: Even / Odd Instructions: ▪ Clearly identify and describe the key concepts needed to develop the program. ▪ Structure the program into separate, reusable modules or functions.					50

Scheme of assessment		
a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10		
b) Implementation and Execution - 30		
c) Best Practices (Code Readability and Error Handling)- 10		
Total Marks		50

Signature of the Course Coordinator

Signature of the HOD

9. SEE–Model Theory Question Paper

Program	Information Science and Engineering	Semester	2
Course Name	Thinking Programming with Python	Marks	50
Course Code	25CS21I	Duration	90 Min

Note: Answer any one full question from each section. Each full question carries equal marks.

Q No	Questions	Cognitive Levels	Course Outcomes	Marks
Section -1				
1	a. Your College is hosting a science fair, and you are responsible for planning and organizing the event. The goal is to ensure the fair runs smoothly, including setting up booths, scheduling presentations, managing participants, and inviting visitors. Use decomposition to divide the problem into smaller tasks.	2	1	
2	a. Five students—Alice, Bob, Charlie, Diana, and Ethan—participated in a coding competition. Based on their scores, the following information is known: 1. Bob scored higher than Diana but lower than Charlie. 2. Alice scored lower than Ethan but higher than Diana. 3. Ethan did not score the highest. 4. Charlie scored the highest. What is the ranking of the students from highest to lowest?	2	1	
Section -2				
3	b. Develop a step-by-step algorithm to check whether a given number is a palindrome. A number is considered a palindrome if it remains the same when its digits are reversed. For example: Input: 1221 → Output: Palindrome Input: 123 → Output: Not a Palindrome Your algorithm should clearly handle positive integers and explain each step, from receiving input to displaying the result.	2	2	
4	a. Design an algorithm to determine the correctness of a One-Time Password (OTP) entered by a	2	2	

	user. The algorithm should: ▪ Prompt the user to input an OTP. ▪ Compare the entered OTP with the pre-stored or system-generated OTP. ▪ Validate whether the OTP matches the expected value. ▪ Provide appropriate feedback for a correct or incorrect OTP. ▪ Allow the user up to three attempts to enter the correct OTP. If the attempts exceed three, lock the process and notify the user. Write the algorithm step-by-step to include all these conditions clearly.			
Section -3				
5	a. Write a program that simulates a simple user authentication system. The program should: 1. Ask the user to input their username and password. 2. Verify the entered credentials against a pre-defined username and password. 3. Allow the user up to three attempts to enter the correct credentials. 4. Provide appropriate feedback for each attempt (e.g., "Incorrect username or password"). 5. If the correct credentials are provided within three attempts, display a success message and terminate the program. 6. If all three attempts are exhausted, display a failure message and terminate the program.	2	3	
6	a. Write a Python program to calculate the factorial of a given number. The program should: 1. Accept a non-negative integer as input. 2. Use the formula $n! = n \times (n-1) \times (n-2) \times \dots \times 1$ 3. Handle invalid inputs (e.g., negative numbers or non-numeric inputs) by displaying appropriate error messages.	2	3	
Section -4				
7	For the below code snippet, identify and fix issues if found. <pre> num = 0 while num != 5: print(f"Current number: {num}") num = num + 2 if num == 5: print("Breaking loop!") break </pre>	3	4	
8	a. Below is a code snippet written to determine whether a number is prime or not. Will the given code	3	4	

	correctly identify whether a number is prime? If not, explain why and suggest how to fix the issue <pre> num = int(input("Enter a number: ")) if num % 2 == 0: print(f"{num} is not a prime number.") else: print(f"{num} is prime number.") </pre>			
Section -5				
9	An online store offers a discount based on the total purchase amount. If the customer's total purchase exceeds a certain threshold, they get a percentage discount. Define a function apply_discount() that accepts the total amount and returns the final price after discount.	3	5	
10	Refactor the following code using a modular approach by breaking it into functions or reusable components. <pre> # Input from user num = int(input("Enter a number: ")) # Check if the number is non-negative if num < 0: print("Factorial is not defined for negative numbers.") else: result = 1 for i in range(1, num + 1): result *= i print(f"The factorial of {num} is result.") </pre>	3	5	

10. Rubrics for Assessment of Activity(Qualitative Assessment)

Sl. No.	Dimension	Excellent (10)	Good (8)	Satisfactory (6)	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
TOTAL						40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computer	i3 or i5	30

SEMESTER 3



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	3
Course Name	Database Concepts & Technologies	Type of Course	Integrated
Course Code	25CS31I	Contact Hours	8 per week
Teaching Scheme	4:0:4	Credits	6
CIE Marks	50	SEE Marks	50 (Theory)

1. Rationale:

This course is designed to equip students with a comprehensive understanding of modern database systems by integrating essential theoretical concepts with practical skills. The course focuses on enabling students to conceptualize, design, and implement both relational and NoSQL databases, with an emphasis on performing basic CRUD (Create, Read, Update, Delete) operations.

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

CO-01	Identify the elements of ER model for a given requirement, draw ER diagram and validate with the given requirement.
CO-02	Translate the ER diagram into a relational schema and verify its compliance with integrity constraints and normalize the design up to the third normal form (3NF)
CO-03	Create database objects and perform data manipulation operations using suitable tool.
CO-04	Query the database to retrieve the required information using suitable SQL statements.
CO-05	Design and implement a NoSQL database, and perform fundamental CRUD operations.

3. Course Content

Week	CO	PO	Lecture(4HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1	Introduction to Database Systems - Overview of databases and their significance - Types of database systems (RDBMS, NoSQL, Distributed, and Cloud Databases.) - Advantages of database systems over file systems - Overview of Database Management Systems (definition, key features and examples)	- Setting up database environments (MySQL/PostgreSQL and MongoDB Atlas) - Explore the interface (MySQL Workbench/ PostgreSQL PgAdmin and local or cloud MongoDB)
2	1	1	Database Models and Architecture a. Database models: ER, Relational, and	a. Consider a database scenario and identify the levels of abstraction (external, conceptual, and internal).

			NoSQL b. Three-tier architecture and DBMS components c. Data Independence: Logical and Physical	b. Analyze a case study where a database underwent schema evolution (e.g., adding/removing tables or columns). Identify how logical independence was maintained. c. Study a database migration example where physical storage was optimized (e.g., moving to SSDs or cloud-based storage) without affecting application functionality. d. Set up database connections (MySQL/PostgreSQL/MongoDB) in Python using appropriate modules.
3	1	1	ER Modeling - Introduction to Entity-Relationship (ER) model - Entities, attributes, relationships, and Constraints.	a. Identify the ER elements for the given requirements. b. Drawing ER diagrams for simple use cases c. Design an ER diagram for a library or e-commerce system. d. Design a comprehensive ER diagram for a healthcare management system, including cardinalities and participation constraints for entities like "Doctor," "Patient," "Appointment," and "Medicine." e. Use database modeling tools like MySQL Workbench/ PostgreSQL PgAdmin to create ER diagrams and validate them against requirements.
4	2		Relational Model and Integrity Constraints - Introduction to Relational Model - Integrity Constraints - Steps for Converting ER Diagrams into Relational Schema	a. Convert ER model to Relational model b. Translate any given ER diagram into Relational schema c. Use MySQL Workbench or PgAdmin to create database objects and apply integrity constraints through the GUI, without writing SQL statements
5	3		Structured Query Language (SQL) – Basics - Introduction to SQL; - Categories of SQL Commands with definition and syntax - Data types in SQL - Update anomalies	Create database, tables and enforce data integrity constraints using DDL commands. Perform basic CRUD operations through programming language such as PYTHON.
6	2		Functional Dependency(FD) - Definition and examples. - Types of functional dependencies - Role of FD in normalization. Normalization - Purpose and benefits of normalization. - Normal forms: 1NF, 2NF, 3NF	- Identify Functional Dependencies in a Table - Consider relations and apply normalization rules to bring them to 1NF, 2NF, and 3NF.

7	4	Querying Databases - SQL Clauses: Where, Distinct, Order By, Limit; - SQL Operators: Logical Operators, Comparison Operators - Aggregate functions	- Write SQL queries to filter, sort, and limit data using clauses. - Use operators to refine queries and match complex conditions. - Apply aggregate functions to summarize data and perform calculations.
8	4	Advanced Database Querying - Joins: INNER, OUTER, CROSS - GROUP BY and HAVING clauses - Subqueries: Single-row subqueries; Multi-row subqueries; Scalar subqueries; Correlated subqueries; - Views	- Perform different types of joins to combine data from multiple tables. - Group data and filter the results using GROUP BY and HAVING clauses. - Write and optimize subqueries for filtering and data retrieval. - Create and use views to simplify complex queries and manage data.
9	3	Transactions and Concurrency Introduction to Transactions Concurrency Control: - Need for Concurrency Control; - Concurrency Issues; - Concurrency Control Techniques; TCL commands DCL commands	- Create, manage, and roll back transactions - Understand and apply locking mechanisms to control concurrent access to data. - Execute DCL commands to provide privileges to different users
10	5	Introduction to NoSQL Databases - Overview; Key Features; Types; Advantages and Disadvantages; - CAP theorem - Comparison between RDBMS and NoSQL - Overview of MongoDB	MongoDB Shell Basic commands : show dbs, use <dbname>, show collections
11	5	MongoDB Data Model and CRUD Operations MongoDB Data Model: Collections and Documents; Data types CRUD Operations - Create: <i>insertOne()</i>, <i>insertMany()</i> - Read: <i>find()</i>, <i>findOne()</i> - Update: <i>updateOne()</i>, <i>updateMany()</i> - Delete: <i>deleteOne()</i>, <i>deleteMany()</i>	- Create MongoDB for E-Commerce Website. - Perform CRUD operations
12	5	Aggregation in MongoDB - Aggregation Pipeline - Stages in the Aggregation Pipeline - Expression Operators	Write the aggregation pipeline
13	3	Database Design and Implementation Developing a mini-project involving both RDBMS and NoSQL databases	Designing and implementing a database for a real-world use case

4. Suggested programs/activities:

Week	Suggested Programs/activities
1	- Setting up database environments (MySQL/PostgreSQL and MongoDB Atlas) - Explore the interface (MySQL Workbench/ PostgreSQL PgAdmin and local or cloud MongoDB)
2	Identifying Levels of Abstraction in a Given Database Scenario Scenario: A retail company uses a database to manage inventory, sales, and customer information. The system allows: - Customers to browse products and place orders through an online shopping platform. - Store managers to track inventory levels and restock products. - The finance team to generate revenue reports and analyze sales trends.
3	Design ER model for the below database requirement. 1. A food delivery platform connects customers with multiple restaurants, allowing them to browse menus, place orders, and have food delivered to their location. The database must manage a wide range of data, including restaurant information, menu items, customer profiles, orders, payment processing, and delivery tracking. The system must also support delivery logistics, managing driver assignments, and tracking order status from preparation to delivery. - Restaurant Management: - Store restaurant details, including name, location, contact information, and operational hours. - Manage menus with categories, items, pricing, and availability status. - Menu Management: - Maintain records of menu items, including ingredients, preparation time, price, and availability. - Support special offers, discounts, and promotions for menu items. - Customer Management: - Store customer profiles with information such as name, contact details, addresses, and order history. - Allow customers to save favorite restaurants, past orders, and payment preferences. - Order Management: - Track customer orders from placement through preparation, dispatch, and delivery. - Manage order status updates in real-time, including stages such as "Preparing," "Out for Delivery," and "Delivered." - Delivery Management: - Assign delivery drivers to orders based on location, availability, and estimated delivery time. - Track driver locations in real-time for efficient route optimization and status updates. - Payment Processing: - Handle various payment methods including credit/debit cards, digital wallets, and cash on delivery. Maintain secure transaction records and support refunds for cancellations or disputes. 2. Design a database for a To-Do List Application that helps users manage their tasks effectively. The database should store and organize information about users, tasks, categories, and their relationships. Requirements: ②The system should store user information such as unique username, email, and password.

	❓ Users should be able to create tasks with attributes like title, description, priority, status((e.g., Pending, In Progress, Completed), and due date. Users should be able to organize tasks into categories. A task can belong to multiple categories, and a category can include multiple tasks. Each task should be linked to the user who created it.																																																
4	Convert above ER Diagram to Relational Schema.																																																
5	Create Retail_Store_Management Database and Product Table with following requirements: - each product should have a unique identifier - Each product should have a Product Name, which must be unique and cannot be NULL - Each product should have a Price, and the price must be greater than zero - Each product should have a Description, whose value can be null. Modify Structure: After creating the table, Remove the Description field from the table structure (as per requirement). - Add Quantity , and the default value for it should be set to 0. Execute DML Commands on the Created Tables - Insert Data into the Product Table - Update Data in the Product Table - Delete Data from the Product Table - Select Data from the Product Table - Examine update anomalies and understand their impact on data integrity Design a database for an Inventory Management System for a retail store. The system should keep track of the store's products, suppliers, and customer orders. The database must include the following requirements: - Each product has a unique ID, name, category, quantity in stock, price per unit, and supplier details. - Each supplier has a unique ID, name, contact information, and the products they supply. - Each customer has a unique ID, name, contact details, and purchase history. Each order must have an order ID, customer details, date of the order, products purchased, quantities, and total cost.																																																
6	Normalize given schema up to 3NF <table><tr><th>OrderID</th><th>CustomerName</th><th>ProductNames</th><th>ProductQuantities</th><th>ProductPrices</th><th>TotalAmount</th><th>ShippingAddress</th></tr><tr><td>1001</td><td>Alice</td><td>Laptop, Mouse</td><td>1, 2</td><td>1000, 50</td><td>1100</td><td>123 Elm St, NY</td></tr><tr><td>1002</td><td>Bob</td><td>Keyboard, Monitor</td><td>1, 1</td><td>80, 300</td><td>380</td><td>456 Maple Ave, CA</td></tr><tr><td>1003</td><td>Charlie</td><td>Laptop, Mouse, Keyboard</td><td>1, 2, 1</td><td>1000, 50, 80</td><td>1130</td><td>789 Birch Rd, TX</td></tr></table> <table><tr><th>MovieID</th><th>MovieTitle</th><th>Cast</th><th>DirectorName</th><th>Genre</th></tr><tr><td>1</td><td>Inception</td><td>Leonardo DiCaprio, Joseph Gordon-Levitt</td><td>Christopher Nolan</td><td>Sci-Fi</td></tr><tr><td>2</td><td>The Dark Knight</td><td>Christian Bale, Heath Ledger</td><td>Christopher Nolan</td><td>Action</td></tr><tr><td>3</td><td>The Prestige</td><td>Christian Bale, Hugh Jackman</td><td>Christopher Nolan</td><td>Drama</td></tr></table>	OrderID	CustomerName	ProductNames	ProductQuantities	ProductPrices	TotalAmount	ShippingAddress	1001	Alice	Laptop, Mouse	1, 2	1000, 50	1100	123 Elm St, NY	1002	Bob	Keyboard, Monitor	1, 1	80, 300	380	456 Maple Ave, CA	1003	Charlie	Laptop, Mouse, Keyboard	1, 2, 1	1000, 50, 80	1130	789 Birch Rd, TX	MovieID	MovieTitle	Cast	DirectorName	Genre	1	Inception	Leonardo DiCaprio, Joseph Gordon-Levitt	Christopher Nolan	Sci-Fi	2	The Dark Knight	Christian Bale, Heath Ledger	Christopher Nolan	Action	3	The Prestige	Christian Bale, Hugh Jackman	Christopher Nolan	Drama
OrderID	CustomerName	ProductNames	ProductQuantities	ProductPrices	TotalAmount	ShippingAddress																																											
1001	Alice	Laptop, Mouse	1, 2	1000, 50	1100	123 Elm St, NY																																											
1002	Bob	Keyboard, Monitor	1, 1	80, 300	380	456 Maple Ave, CA																																											
1003	Charlie	Laptop, Mouse, Keyboard	1, 2, 1	1000, 50, 80	1130	789 Birch Rd, TX																																											
MovieID	MovieTitle	Cast	DirectorName	Genre																																													
1	Inception	Leonardo DiCaprio, Joseph Gordon-Levitt	Christopher Nolan	Sci-Fi																																													
2	The Dark Knight	Christian Bale, Heath Ledger	Christopher Nolan	Action																																													
3	The Prestige	Christian Bale, Hugh Jackman	Christopher Nolan	Drama																																													

	<table><tr><th>BookID</th><th>BookTitle</th><th>Authors</th><th>PublisherName</th><th>AuthorContact</th></tr><tr><td>1</td><td>Python Programming</td><td>John Doe, Jane Smith</td><td>TechBooks</td><td>john@tech.com, jane@tech.com</td></tr><tr><td>2</td><td>Data Science Basics</td><td>Alice Brown, Bob White</td><td>DataPress</td><td>alice@datasci.com, bob@datasci.com</td></tr><tr><td>3</td><td>Web Development</td><td>Alice Brown, John Doe</td><td>WebWorks</td><td>alice@webworks.com, john@webworks.com</td></tr></table> <table><tr><th>OrderID</th><th>CustomerName</th><th>ProductNames</th><th>ProductPrices</th><th>ShippingAddress</th></tr><tr><td>101</td><td>Alice</td><td>Laptop, Mouse</td><td>1000, 50</td><td>123 Elm St, NY</td></tr><tr><td>102</td><td>Bob</td><td>Keyboard, Monitor</td><td>80, 300</td><td>456 Maple Ave, CA</td></tr><tr><td>103</td><td>Charlie</td><td>Laptop, Mouse, Keyboard</td><td>1000, 50, 80</td><td>789 Birch Rd, TX</td></tr></table>	BookID	BookTitle	Authors	PublisherName	AuthorContact	1	Python Programming	John Doe, Jane Smith	TechBooks	john@tech.com , jane@tech.com	2	Data Science Basics	Alice Brown, Bob White	DataPress	alice@datasci.com , bob@datasci.com	3	Web Development	Alice Brown, John Doe	WebWorks	alice@webworks.com , john@webworks.com	OrderID	CustomerName	ProductNames	ProductPrices	ShippingAddress	101	Alice	Laptop, Mouse	1000, 50	123 Elm St, NY	102	Bob	Keyboard, Monitor	80, 300	456 Maple Ave, CA	103	Charlie	Laptop, Mouse, Keyboard	1000, 50, 80	789 Birch Rd, TX
BookID	BookTitle	Authors	PublisherName	AuthorContact																																					
1	Python Programming	John Doe, Jane Smith	TechBooks	john@tech.com , jane@tech.com																																					
2	Data Science Basics	Alice Brown, Bob White	DataPress	alice@datasci.com , bob@datasci.com																																					
3	Web Development	Alice Brown, John Doe	WebWorks	alice@webworks.com , john@webworks.com																																					
OrderID	CustomerName	ProductNames	ProductPrices	ShippingAddress																																					
101	Alice	Laptop, Mouse	1000, 50	123 Elm St, NY																																					
102	Bob	Keyboard, Monitor	80, 300	456 Maple Ave, CA																																					
103	Charlie	Laptop, Mouse, Keyboard	1000, 50, 80	789 Birch Rd, TX																																					
7	Suggestive Basic SQL Queries on To do list database - Retrieve User Information for specific email id - Count Tasks for all Users - Retrieve the First 5 Tasks - Retrieve Tasks Due Today - Find the Maximum Due Date of Tasks for a Specific User - Count the Number of Pending Tasks for a Specific User - Find the Average Number of Tasks for specific user																																								
8	Suggestive Advanced SQL Queries on To do list database - Retrieve Tasks with No Categories Assigned. - Count Tasks in Each Category. - Retrieve Categories Without Any Tasks - Retrieve Users Who Have Tasks with High Priority - Retrieve Tasks and Their Categories using JOIN - Retrieve Categories with More Than 5 Tasks Assigned. - Retrieve Users with Their Total Task Count using subquery																																								
9	Create a transaction for the scenario: - A customer wants to transfer money from their savings account to their checking account. - If any of the steps fail (for example, if there is insufficient balance in the savings account), roll back the transaction to ensure that no changes are made to the database. DCL - Grant SELECT privilege on a table to a user - Grant multiple privileges to a user - Grant all privileges on a table to a user - Grant privileges on all tables in a schema - Revoke SELECT privilege from a user - Revoke multiple privileges from a user - Revoke all privileges from a user - Revoke privileges on all tables in a schema																																								
10	MongoDB Shell Basic commands : show dbs, use <dbname>, show collections																																								

11	Create MongoDB for E-Commerce Website with following collections Products: Stores product details like name, description, price, category, etc. Users: Stores user details like username, email, password, etc Orders: Stores order details like user ID, product IDs, order status, etc. ▪ Adding a Product ▪ Adding a User ▪ Adding an Order ▪ Find All Products in a Category ▪ Find a Specific Product by Name ▪ Retrieve Orders for a Specific User ▪ Find User Details by Username ▪ Update Product Stock ▪ Update User Address ▪ Delete a Product ▪ Delete an Order
12	Write the aggregation pipeline ▪ to calculate the total sales for each product within a specified date range. ▪ to calculate the average price of products in each category ▪ to filter products within a certain price range and then calculate the total number of products in each category.

5. CIE Assessment Methodologies

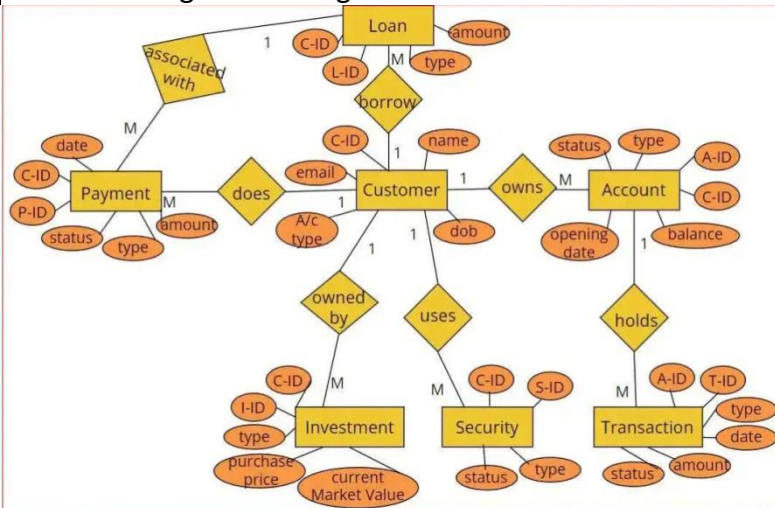
Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

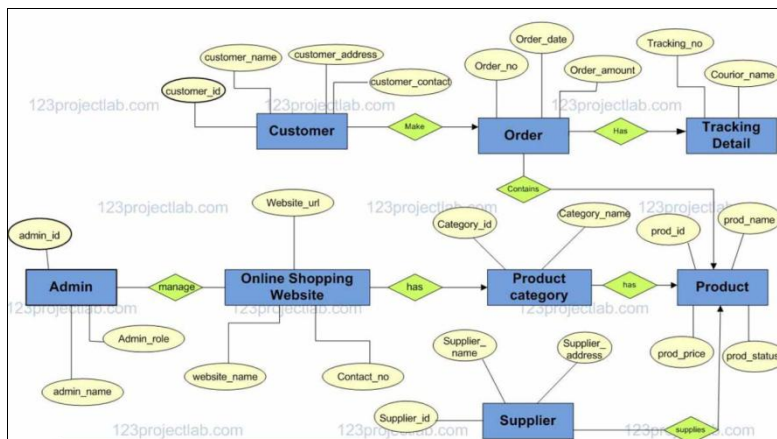
6. SEE - Theory Assessment Methodologies

Sl.No	SEE – Theory Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Theory	90	50	20

7. CIE Theory Test model question paper

Program	Information Science and Engineering	Semester -3	
Course Name	Database Concepts & Technologies	Test	I/III

Course Code	25CS31I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	Questions	Cognitive Level	Course Outcome	Marks	
Section - 1					
1	a. Why is data integrity better maintained in databases than in file systems? Provide examples. – 5 b. Identify and explain the key features of a DBMS. – 10 c. Explain the difference between total and partial participation with examples. – 4 d. Identifying Levels of Abstraction in below given Database Scenario - 6 A university uses a database to manage information about students, courses, and enrollments. The database allows: - Students to view their grades and register for courses via an online portal. - Professors to input grades and access class rosters. - The administration to generate reports on enrollment statistics and course performance.	2,3	1	25	
2	a. Describe the three-tier architecture of a DBMS and explain the role of each tier. - 5 b. Create an ER diagram for a retail store that includes entities like "Product," "Customer," "Order," and "Supplier." Include attributes and relationships and constraints. - 20	2,3	1		
Section - 2					
3	Translate the given ER diagram to relational schema 	3	2	25	
4	Translate the given ER diagram to relational schema	3	2		



Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.

Sign of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	3
Course Name	Database Concepts & Technologies			Test	II/IV
Course Code	25CS31I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
Develop a database system to support a Ticket Raising Application that facilitates efficient management and resolution of user-reported issues. The system should be capable of handling user information, ticket details, categorization, prioritization, assignment, and tracking of ticket resolution. Requirements: - Store user information such as name, email, phone number, and role (e.g., Customer, Support Agent, Admin). - Maintain secure login credentials for users. - Allow customers to raise tickets with details such as: - Issue description - Category (e.g., Technical, Billing, General Inquiry) - Priority level (e.g., Low, Medium, High, Critical) - Assign a unique identifier and timestamp to each ticket. - Track the status of each ticket (e.g., Open, In Progress, Resolved, Closed) and record timestamps for status changes. - Store resolution details, including: - Comments or actions taken by the support agent and resolution timestamp. - Assign tickets to support agents based on expertise, workload, or availability.					50
Scheme of assessment					
a. Database Design – 15 - Entity-Relationship (ER) Diagram -5 - Correct conversion of ER diagram into a normalized relational schema. - 5 - Use of appropriate data types for attributes and Implementation of integrity constraints-5 b. Database Implementation – 20 - Correct creation of database objects using tools - 5 - Application of constraints - 5 - Data Population - 5 - Execution of basic INSERT, UPDATE, DELETE, and SELECT statements - 5 c. Querying (basic and advanced) - 10 d. Documentation and Presentation - 5					
Total Marks					50

Sign of the Course Coordinator

Signature of the HOD

9. SEE–Model Theory Question Paper

Program	Information Science and Engineering	Semester	3	
Course Name	Database Concepts & Technologies	Marks	50	
Course Code	25CS31I	Duration	90 Min	
Note: Answer any one full question from each section. Each full question carries equal marks.				
Q No	Questions	Cognitive Levels	Course Outcomes	Marks
Section -1				

1	a. Explain how the components of a DBMS work together to handle a query. b. Given a description of a university database, draw an ER diagram with entities, attributes, and relationships.	2	1	10
2	a. Identify entities, attributes, and relationships for a hospital management system. b. Explain the differences between the external, conceptual, and internal schemas with examples.	2	1	10
Section -2				
3	a. Write a relational schema for a library management system with at least 5 attributes and specify the keys and constraints. b. Normalize the following table to 3NF: OrderID CustomerName ProductName SupplierName SupplierContact Identify the keys and explain each step.	2,3	2	10
4	a. Explain the integrity constraints in the context of database b. Identify the type of FD in the following example: If EmployeeID determines Department, and Department determines Manager, then EmployeeID determines Manager.	2,3	2	10
Section -3				
5	a. What are the five categories of SQL commands? Provide one example for each. - 5 b. Explain the use of COMMIT and ROLLBACK in an inventory management system with an example. - 5	2	3	10
6	a. What are the ACID properties of a transaction? Provide examples for each property. – 5 b. Write an SQL query to create a table Employees with columns: EmpID (Primary Key), Name (Not Null), Salary (Decimal), and Department (Varchar). - 5	2	3	10
Section -4				
7	a) Write an SQL query to retrieve records from the "inventory" table where "stock" is less than 10 but not equal to 0. b) Write an SQL query to find all orders where the "quantity" is between 10 and 50 or the "status" is "shipped". c) Write an SQL query to count the number of employees in each department.	3	4	10
8	a) Write an SQL query to sort the "products" table by price in descending order and display only the first 10 results.	3	4	10

	b) Write an SQL query to find the average price of products in the "products" table. c) Write a single-row subquery to find the product with the highest price.			
Section -5				
9	a. Explain the CAP theorem with examples. – 6 b. Explain the difference between - 4 1. insertOne() and insertMany() 2. find() and findOne()	3,2	5	10
10	a. Given a collection of student records, write queries to: 1. Insert multiple student records. 2. Find students with marks greater than 80. 3. Update the "status" field for students with marks below 50 to "at risk". - 6 b. What is the purpose of the aggregation framework in MongoDB? – 4	3, 2	5	10

10. Rubrics for Assessment of Activity (Qualitative Assessment)

Sl. No.	Dimension	Excellent	Good	Satisfactory	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
TOTAL						40

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	i3/i5 processor	30
02	Internet connection	100mbps	



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	3
Course Name	Computer Networks	Type of Course	Integrated
Course Code	25CS32I	Contact Hours	8 per week
Teaching Scheme	4: 0:4	Credits	6
CIE Marks	50	SEE Marks	50 (Theory)

1. Rationale:

This course is designed to equip students with the foundational knowledge and practical skills needed to design, implement, and troubleshoot computer networks in various environments. It addresses the selection of appropriate communication technologies and the resolution of issues like signal loss, latency, and noise to ensure optimal network performance. Learners will gain hands-on experience in configuring IP addressing, routing, subnetting, and protocols like TCP/UDP and DNS using network simulators. The course also emphasizes designing secure and scalable networks, from home office setups to enterprise-level solutions, tailored to specific requirements. By mastering these skills, students will be well-prepared for roles in network administration, engineering, and IT consulting.

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

CO-01	Select an appropriate communication technology for a given network and ensure optimal performance by addressing issues arising from transmission impairments
CO-02	Design a network for a given specification by using the right network components, devices, topologies, protocols and software.
CO-03	Troubleshoot common network issues.
CO-04	Demonstrate the configuration of IP address, routing, subnetting, client-server interaction (TCP, UDP) and DNS for a given WAN network using a network simulator.
CO-05	Design and implement simple Home Office network for given specification.

3. Course Content

Week	CO	PO	Lecture(4HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1,2	1,7	Perspectives on Networking – End user perspectives on Networking and Internet. Communication Systems - Basic elements of communication systems with block diagram, List commonly used terms in electronic communication systems. Data representation, Data flow. Communication Channels: Guided Media (Over view)- Twisted pair, Coaxial Cable, Optical Fibers. Analog and Digital Signals, Periodic and Non-Periodic Signals, Sine Wave,	1. Demonstrate different network cables and connectors. 2. Demonstrate connecting Computer to home modem/router through Ethernet cable or Wifi 3. Automatic and Manual IP Setting (Computer and Smart phone) and demonstrate connecting to Internet, discuss IP address, Subnet mask, Default gateway and DNS in end user perspective.

			Phase, Wavelength, Digital Signals, Bitrate, Bit-length, Modulation, Demodulation. Unguided Media: Micro Wave, Radio wave, Infrared etc. Electromagnetic waves – what are electromagnetic waves and their properties Electromagnetic spectrum - classification and its applications.	
2	1,2,3	1,2,7	Transmission Impairment – Attenuation, Distortion and Noise, Performance - Bandwidth, Throughput, Latency, Jitter (Basic concepts only). Transmission Modes – Parallel and Serial Transmission. Asynchronous and Synchronous Transmission. Overview of Networking. Categories of networks – Personal Network, LAN, MAN, WAN, Internetworking (Illustrate Network from LAN connected using a HUB to Internetwork). A brief History – ARPANET, MILNET, CSNET, NSFNET, ANSNET. The Internet today – Backbone ISP, Regional ISP, Local ISP. World Wide Web.	1. Explore all ISP in your area/locality and select best internet ISP/plan based on cost and performance. 2. Test the download/upload speed in your computer/mobile phone also check type, bandwidth and ISP. 3. Demonstrate Transmission mode (Video Demonstration) 4. Demonstrate Bandwidth, Throughput, Latency, Jitter (Video Demonstration) 5. Use pathping command to find actual path between source to destination with information about network latency/delay & network loss.
3	1,2,3	1,2,3,7	Protocol and Standards- The communication rules (Method, language, Confirmation) – Protocols. Standards. Protocol layers OSI Model : OSI Layers and Their Functions, OSI Layering Concepts and Benefits, OSI Encapsulation Terminology.	Network Simulator: Demonstrate Packet tracer or any other network simulator 1. Installation 2. User Interface 3. Components and cables 4. GUI and CLI Configuration 5. Configure end devices. 6. Demonstrate Working of all 7 layers by opening a webpage in a simulated network using packet tracer
4	1,2,3	1,2,3,7	TCP/IP Networking Model - Overview of the TCP/IP Networking Model. TCP/IP Application Layer, HTTP Overview, HTTP Protocol Mechanisms. TCP/IP Transport Layer, TCP Error Recovery Basics, Same-Layer and Adjacent-Layer Interactions. TCP/IP Network Layer, Internet	1. Determine the IP Address Configuration of a Computer (Windows) and Test the Network Interface TCP/IP Stack (Ping). 2. Install and configure wireless access point over the LAN. 3. Determine the MAC Address of a Host (PC and Phone). 4. View Wireless and Wired NIC

			Protocol and the Postal Service, Internet Protocol Addressing Basics, IP Routing Basics. TCP/IP Link Layer (Data Link Plus Physical), TCP/IP Model and Terminology, Data Encapsulation Terminology. Names of TCP/IP Messages. Addressing : Physical, Logical, Port	Information and make a table explaining each. 5. Manual and Automatic address assignment (Windows and Android Pone) a) IPv4 address b) Subnet mask c) DNS 6. Demonstrate Addressing in packet tracer
5	1,2,3,5	1,2,3,4	Hardware and Software components of Network - Common network devices - Computers, Access points, Hub, Switch, Router, repeaters, NIC, Modem. LAN Cables – Co-axial, twisted pair, optical fibre, LAN connectors- co-axial cable, and twisted pair cable, optical fibre, Connectors, Firm wares, ISPs. Wired Local Area Networks Protocol and technologies: Ethernet (IEEE 802.3) – Frame Format, Ethernet Evolution, Access Methods: CSMA/CD, CSMA/CA, Fast Ethernet, Gigabit Ethernet. Error Detection in Ethernet. Broadcast and Collision domain. Overview of network topologies - Basic topologies- bus, ring, star, mesh and hybrid.	1. Demonstrate working of common network devices. 2. Install and configure NIC. 3. Twisted Pairs: UTP Cabling Pinouts for 10BASE-T and 100BASE-T, Straight Through Cable Pinout, Crossover Cable Pinout Demonstration. 4. Crimping of RJ45: Straight and Cross. a) Punching Cat 6 cable to I/O Box. Use punching tool. b) Check connectivity using LAN tester. 5. Create and Demonstrate all possible network topologies using simulator. 6. Demonstrate Broadcast and Collision domain (Video demonstration)
6	1,2,3,5	1,3,4	Wireless LAN (IEEE 802.11) – Architecture, Frame Format,BSS, ESS, MAC Sublayer, CSMA/CA, Hidden and Exposed Station Problems, Ethernet access layer devices – Hub, Switch, The MAC address table, Ethernet Broadcast and Broadcast domain, ARP. Hierarchical Network Design – Physical and logical addresses, Benefits of a Hierarchical Design, Access, Distribution, and Core layers (Demonstrate Hierarchical Network Design in packet tracer)	1. Build a physical Ethernet LAN Network and demonstrate file sharing, printer sharing. 2. Install and configure wireless access point over the LAN. 3. Configure and install a ethernet switch/Hub (Use simulator if hardware devices are not available) 4. Create/model a simple Ethernet network using 3 hosts and a switch, Observe traffic behavior on the network and Observer data flow of ARP broadcasts and pings. 5. Demonstrate Collision and Broadcast domain in packet tracer.
7	2,4,5	1,3,4	Introduction to Network Layer	1. Build a simple peer-to peer

			Switching - Circuit Switching, Packet Switching. Packet Switching in Network Layer- Connectionless Service, Connection-Oriented Service. Network Layer Services – Logical Addressing, Services Provided at the Source Computer, Services Provided at Each Router, Services Provided at the Destination Computer. Other Network Layer Issues - Error Control, Flow Control, Congestion Control, Routing	network and verify physical connectivity and Assign various IPv4 addresses to hosts and observe the effects on network communication. 2. Configure IP addresses of a network (real or simulated) and ping across to test and troubleshoot. 3. View Wireless and Wired NIC Information and make a table explaining each.
8	2,4,5	1,3,4	IP Addressing, Address Space, Notation. Classful addressing , Two Levelling addressing, Three-Level Addressing, Classless addressing – Subnetting, Subnet mask, Super netting, CIDR, Variable length subnet mask. Special addresses Network Address Translation (NAT). Delivery and forwarding of IP packets - Direct Delivery, Indirect Delivery. Forwarding. Structure of a Router. IP V4 – Datagram, fragmentation.	1. Subnetting : Your college is given an ip 192.168.100.0/24, you have 5 different departments, design and implement subnetting using packet tracer. 2. Connect to web server using simulator, Demonstrate how packets are sent across the Internet using IP addresses and working of HTTP services. 3. Implement Network Address Translation using packet tracer
9	2,4,5	1,3,4	Routing: The Need for Routing - Criteria for Dividing the Local Network. IPv4 Routing - IPv4 Host Routing, Router, Forwarding Decisions and the IP Routing Table, The default gateway, A Summary of Router Forwarding Logic, A Detailed Routing Example. Routing Protocols - IPv4 Routing Protocols – static and dynamic. Other Network Layer Features - Using Names and the Domain Name System, The Address Resolution Protocol, ICMP Echo and the ping Command. IGMP. IPv4 Issues - Need of IPv6, Ipv4 vs IPv6, IPv4 and IPv6 Coexistence. IPv6 features - IPv6 Address Representation. Concept of Virtual LAN's(VLAN's)	1. Implement simple static routing for the practice activity of week 8 (subnetting) 2. Troubleshooting of IP Addressing a) Change a routing table entry b) Wrong address c) incorrect subnet mask. 3. Demonstrate ICMP commands.
10	2,4,5	1,3,4	Transport Layer Services- Process-to-Process Communication, Addressing: Port Numbers, Multiplexing and	1. Demonstration of Port addressing and well-known port numbers (HTTP, DNS, SMTP) using packet

			Demultiplexing, Flow Control, Error Control – Sliding window. Congestion Control. Connectionless and Connection-Oriented Services. Transport Layer protocols- Simple protocol, Stop and Wait, Go back N.	tracer - TCP vs. UDP Comparison (Compare the behavior of TCP and UDP protocols) - Create a network with two PCs connected through a router. - Configure one PC to run an HTTP (TCP) server and the other a DNS (UDP) server. - Use the Web Browser and DNS Client tools on the other PC to access these services. - In Simulation Mode, filter for TCP and UDP packets and compare reliability, acknowledgment, and retransmission behavior. - Demonstrate Sliding window, Congestion control, Stop and Wait, Go back N (Only demonstration)
11	2,4,5	1,3,4	The Client Server relationship – Client Server interaction. URI, URN, URL User datagram protocol – User datagram, UDP services, UDP Applications. Transmission Control protocol: TCP services , features. Popular TCP/IP Applications, Connection Establishment and Termination. Port Numbers – TCP and UDP. Socket pairs, The netstat command.	- Create a client – server model in simulator and observe the client interaction between the server and PC using packet tracer. - Create a network with one server, a switch, and two PCs. - Configure IP addresses for all devices. - Enable HTTP and FTP services on the server. - Use the Web Browser and Command Prompt on the clients to access HTTP and FTP services. - Open Simulation Mode to capture and analyze HTTP GET and FTP request/response packets. - Demonstrate Netstat command
12	2,4,5	1,3,4	Introduction to application layer: Client Server paradigm. HTTP DHCP DNP Telnet and SMTP,POP, IMAP, FTP.	- Configure a DNS server using packet tracer and Observe the conversion of a URL to an IP address. - Observe DNS lookup using the nslookup command. - Use simulator to demonstrate Telnet and SSH - Configure and test DHCP using

				packet tracer. 5. Configure a HTTP service and demonstrate working of HTTP
13	1,2,3	1,2,3,4,7	Troubleshoot Common Network Problems - The Troubleshooting Process, Network Troubleshooting Overview, Gather Information - Nature of problem, Equipment, Configuration and Topology, Previous Troubleshooting. Structured Troubleshooting Methods - Bottom-Up, Top-Down, Divide-and-Conquer, Follow-the-Path, Substitution, Comparison, Educated Guess. Guidelines for Selecting a Troubleshooting Method Troubleshoot Wireless Issues - Causes of Wireless Issues, Authentication and Association Errors.	1. Demonstrate troubleshooting Commands with a scenario- ipconfig, ping, netstat, tracert, nslookup. 2. Troubleshooting Common Issues Using Packet Tracer Simulation Mode a) set up a basic network with a router, switch, and PCs. b) Introduce issues like: c) Incorrect subnet mask. d) Disabled router interfaces. e) Use Simulation Mode to observe packet flow: f) Identify where packets drop or fail. g) Fix the configuration and test.

4. Learning Resources

Week	Resources
1	1. NCERT Class 12 Physics Chapter 8 2. Networking courses on Netcad.com : Networking basics and Networking essentials 3. Skillsforall.com : Networking basics and Networking essentials 4. Youtube.com 5. ChatGPT
2	1. NCERT Class 12 Computer Science 2. TCP/IP protocol suite Behrouz A. Forouzan 3. Data Communication and Networking Behrouz A. Forouzan 4. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 5. Computer Networks by Andrew S. Tanenbaum 6. Networking courses on Netcad.com : Networking basics and Networking essentials 7. Skillsforall.com : Networking basics and Networking essentials
3	1. NCERT Class 12 Computer Science 2. TCP/IP protocol suite Behrouz A. Forouzan 3. Data Communication and Networking Behrouz A. Forouzan 4. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 5. Computer Networks by Andrew S. Tanenbaum 6. Networking courses on Netcad.com: Networking basics and Networking essentials, Getting Started with Cisco Packet Tracer 7. Skillsforall.com: Networking basics and Networking essentials
4	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com : Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials

5	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials, Getting Started with Cisco Packet Tracer 6. Skillsforall.com: Networking basics and Networking essentials, Getting Started with Cisco Packet Tracer 7. Cisco.com : Connector and Cable Specifications
6	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials, Getting Started with Cisco Packet Tracer 6. Skillsforall.com: Networking basics and Networking essentials, Getting Started with Cisco Packet Tracer
7	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials
8	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials 7. https://www.youtube.com/watch?v= BVvXph43aM 8. https://techacadhelp.com/11-5-5-packet-tracer-subnet-an-ipv4-network/ 9. https://www.geeksforgeeks.org/static-nat-configuration-in-cisco/
09	1. CP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials 7. https://www.youtube.com/watch?v=ziDv9esbEG0
10	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials 7. https://www.youtube.com/watch?v=D4Re_gv3rz0&t=18s
11	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum

	5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials
12	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials
13	1. TCP/IP protocol suite Behrouz A. Forouzan 2. Data Communication and Networking Behrouz A. Forouzan 3. Computer Networks: A Systems Approach by Bruce S. Davie and Larry L. Peterson 4. Computer Networks by Andrew S. Tanenbaum 5. Networking courses on Netcad.com: Networking basics and Networking essentials 6. Skillsforall.com: Networking basics and Networking essentials

5. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	Average of all CIE=50 Marks
Total					50 Marks

6. SEE - Theory Assessment Methodologies

Sl. No	SEE – Theory Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Theory	90	50	20

7. CIE Theory Test model question paper

Program	Information Science and Engineering			Semester -3	
Course Name	Computer Networks			Test	I
Course Code	25CS321	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					

Q.No	Questions	Cognitive Level	Course Outcome	Marks
Section - 1				
1	a) Illustrate Communication system with block diagram. 5 Marks	L2	1	25
	b) Discuss optical fibers, Why are optical fibers preferred for long-distance communication? 10 Marks	L3	1	
	c) Why is serial transmission preferred for long-distance communication? 10 Marks	L3	1	
2	a) How does data represented in computer? 5 Marks	L3	1	
	b) Why is line-of-sight important for microwave communication? 5 Marks	L3	1	
	c) What is bitrate and bit length? How they are connected to network speed? 5 Marks	L3	1	
	d) Illustrate different categories of networks. 10 Marks	L2	1	
Section – 2				
3	a) Illustrate the seven layers of the OSI model. 10 Marks	L2	2	25
	b) What is encapsulation in the context of the OSI model? Why is encapsulation essential for data transmission in a network? 10 Marks	L2	2	
	c) Illustrate main functions of Network layer in OSI model. 5 marks	L2	2	
4	a) Illustrate working of TCP/IP with diagram 10 Marks	L3	2	
	b) Differentiate between Physical, Logical, Port addressing 10 Marks	L3	2	
	c) Discuss the purpose of routing in the network layer 5 marks	L3	2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	3
Course Name	Computer Networks			Test	IV
Course Code	25CS32I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
Your college is given an IP 192.168.100.0/24, you have 5 different departments and					50

each department needs separate subnetwork, design and implement the network using packet tracer.	2,5	
Or Design a network with minimum 2 routers and 4 switches, configure network and demonstrate static routing using packet tracer.		
Scheme of assessment		
a) Design IP Address scheme (Written)		15
b) Network design (Creating topology, configuring devices, IP assignment etc)		20
c) Verify connectivity / Routing		10
d) Demonstrate working / Troubleshoot if any error		05
Total Marks		50

Signature of the Course Coordinator

Signature of the HOD

9. SEE–Model Theory Question Paper

3. SEE Model Theory Question Paper

Program	Information Science and Engineering	Semester	3
Course Name	Computer Networks	Marks	50
Course Code	25CS32I	Duration	90 Min

Note: Answer any one full question from each section. Each full question carries equal marks.

Q No	Questions	Cognitive Levels	Course Outcomes	Marks
Section -1				
1	Discuss optical fibers, Why are optical fibers preferred for long-distance communication? 10 Marks	L3	1	10
2	a) Why is line-of-sight important for microwave communication? 5 Marks	L3	1	10
	c) What is bitrate and bit length? How they are connected to network speed? 5 Marks	L3	1	
Section -2				
3	What is encapsulation in the context of the OSI model? Why is encapsulation essential for data transmission in a network? 10 Marks	L2	2	10
4	Differentiate between Physical, Logical, Port addressing	L3	2	10
Section -3				
5	a) There are multiple computers are accessing single medium. You are designing a network want to avoid any collision among data packets. How can you achieve it? Explain briefly (CSMA/CA)	L3	5	5
	b) For every Computer there is a Physical address and a logical address why?	L3	5	5
6	a) Differentiate between Circuit Switching and Packet Switching.	L2	4	
	b) Why routing is necessary?	L3	4	

Section -4				
7	a) What is Network Address Translation? Explain	L2	4	5
	b) Your institute allocated with an IP 192.168.100.0/24, you have 5 different departments, design subnetting	L3	4	5
8	a) Change the following IP addresses from dotted-decimal notation to binary notation:	L3	4	2.5+2.5 =5
	a. 114.34.2.8	L3	4	
	b. 129.14.6.8			
	b) Why we are using Sliding window protocol? Explain			5
Section -5				
9	a) Differentiate between Top down and Bottom up approaches in trouble shooting	L3	3	5
	b) How do you thoubleshoot wireless connectivity issues	L2	3	5
10	a) Explain any 3 application layer protocols	L2	2,5	5
	b) The cat-6 cable is connected from computer to outside internet through a switch and a modem, but there is an error saying “No internet”. What are the possible causes for this error? How can you resolve this issue?	L3	3	5

10. Rubrics for Assessment of Activity(Qualitative Assessment)

Sl. No	Dimension	Excellent	Good	Satisfactory	Needs improvement	Score
		9-10	7-8	5-6	Below 5	
1	Knowledge of Hardware Components	Demonstrates an in-depth understanding of all hardware components, their roles, and interconnections with accurate terminology.	Understands most hardware components and their roles, with minor gaps or inaccuracies.	Basic understanding of hardware components but struggles with details or technical terminology.	Limited understanding of hardware components and their functions.	8
2	Configuration and Installation Skills	Configures and installs hardware/software accurately without assistance, following industry best practices.	Configures and installs hardware/software correctly with minimal guidance; few errors.	Completes configuration and installation with frequent guidance; some errors in execution.	Unable to complete configuration or installation without significant errors or assistance.	6
3	Troubleshooting and Problem-Solving	Identifies and resolves issues systematically and efficiently using appropriate tools and methods.	Diagnoses and resolves most issues with minor inefficiencies or errors	Can diagnose and resolve basic issues but struggles with more complex problems.	Unable to diagnose or resolve issues without significant assistance.	2
4	Documentation and Reporting	Produces detailed, well-organized, and professional documentation with	Documentation is clear and organized but contains minor	Documentation is basic, with several errors or incomplete	Documentation is incomplete, unclear, or missing	3

		no errors.	errors or omissions.	details.	important details.	
5	Teamwork and Collaboration	Actively contributes to team efforts, communicates effectively, and assumes leadership when necessary.	Contributes effectively to the team with good communication; may rely on others for leadership.	Participates in team activities but lacks initiative or effective communication.	Limited participation or ineffective collaboration in team tasks.	2
	Total/50					21/50

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers		30
02	Internet Connectivity		



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	3
Course Name	Data structures with Python	Type of Course	Integrated
Course Code	25IS33I	Contact Hours	7 Hours per week
Teaching Scheme	3:0:4	Credits	5
CIE Marks	50	SEE Marks	50 (Practice)

1. Rationale:

The course is designed to provide learners with a comprehensive understanding of how data is organized, stored, and manipulated in computational systems. By covering both linear and nonlinear data structures, as well as Python's built-in data structures, the course equips students with the knowledge and practical skills necessary to design efficient algorithms and solve computational problems effectively.

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

CO-01	Analyze and distinguish between different data structures and their properties to determine their suitability for the use cases.
CO-02	Devise a solution to the given problem by selecting the appropriate built in data structures
CO-03	Develop and implement the solution incorporating error handling mechanisms.
CO-04	Explain the functionality of Abstract Data Types (ADT) such as Stacks, Queues, Lists, and Trees.
CO-05	Implement Abstract Data Types(ADT) using User-defined data structures(UDT) to develop solutions for a given problem.

3. Course Content

Week	CO	PO	Theory	Practice
1	1	1,7	Introduction to Data Structures Classification: Built in data structures User defined data structures Linear & non linear Operations on data structures: Traversal, Insertion, Deletion, Searching, Sorting, and Merging.	Recap of python concepts [practice programs using functions] Explore and document real time applications of Built-in data structures and user defined data structures.
2	1,2	2,4	Sets: - Characteristics and uniqueness of elements. - Set operations: Union, intersection, difference, symmetric difference. - Removing duplicates from a list using sets. Tuples:	Practice programs to understand the concept of - Sets and its operations. - Tuple and its operations

			• Characteristics and immutability. • Basic operations: Creating, Accessing elements and slicing • Packing and unpacking tuples.	
3	1,2	2,4	Lists: • Characteristics and use cases. • Common operations: Creating, Indexing, slicing, appending, removing, sorting. • List comprehensions. Dictionaries: • Key-value pairs: Characteristics and use cases. • Adding, updating, deleting entries. • Iterating through dictionaries. • Dictionary methods (e.g., <code>keys()</code>, <code>values()</code>, <code>items()</code>)	Practice programs to understand the concept of • Lists and its operations. • Dictionaries and its operations
4	1,2	2,4	Strings: • Creating and assigning strings • Indexing String manipulation & built-in methods: • Concatenation • Repetition • Slicing Files: File attributes; operations	Practice programs to understand the concept of • Strings and its methods. • Files and its operations
5	1	2,4	Error and Exception Handling: Types of errors; Exception handling Raising exception	Practice programs to understand the concept of • Exception handling
6	3	2,4	Introduction to Recursion • Definition of recursion. • How recursion works (stack memory and execution flow). • Difference between recursion and iteration. Structure of a Recursive Function • Base case. • Recursive case. • Identifying recursive patterns in problems. Use cases of recursion: Factorial calculation.	Practice programs to understand the concept of recursion
7	1,2,3	2,4	Getting started with ADT and UDT: Abstract data type(ADT): Common ADTs: Stacks, Queues, Lists, Trees, Graphs, Hash Tables. User-Defined Data Type(UDT): class Using UDTs to Implement ADTs	Practice programs to understand the concept of ADT and UDT.
8	1,3	2,4	Stack data structure LIFO (Last In, First Out) principle Real life examples Stack Operations	Practice programs to understand the concept of Stack data structure and its operations

			Implementation of Stack	
9	1,3	2,4	Queue Data Structure FIFO(First In, First Out) principle Real life examples Queue operations Implementation of Queue	Practice programs to understand the concept of Queue data structure and its operations
10	1,3,4	2,4	Linked Lists - Definition and basic structure (Node, Head, Tail) - Linked list vs. arrays - Types of linked lists: Singly Linked List, Doubly Linked List, Circular Linked List Singly Linked List - Structure and representation - Basic operations: - Insertion (at beginning, end, and middle) - Deletion (from beginning, end, and middle) - Traversal - Searching	Practice programs to understand the concept of singly linked list its operations
11	1,3,4	2,4	Doubly Linked List - Structure and representation (with previous and next pointers) - Basic operations: - Insertion (at beginning, end, and middle) - Deletion (from beginning, end, and middle) - Traversal (both forward and backward)	Practice programs to understand the concept of singly linked list and its operations
12	1,3,5	2,4	Introduction to Trees - Definition and Terminology - Properties of Trees - Binary tree representation in memory Basic Operations: - Insertion, Deletion, Searching - Traversal: In-order, Pre-order, Post-order	Practice programs to understand the concept of tree data structure and its operations
13	1,3,5	2,4	Tree Traversal(recursive and iterative): - In-order - Pre-order - Post-order	Practice programs to understand the concept of tree traversals

4. Learning Resources:

Sl No	References
1	"Python for Everybody" by Dr. Charles Severance
2	"Automate the Boring Stuff with Python" by Al Sweigart
3	"Python Data Structures and Algorithms" by Benjamin Baka
4	https://docs.python.org/3/tutorial/datastructures.html
5	https://www.geeksforgeeks.org

6	https://www.w3schools.com
7	https://www.tutorialspoint.com
8	https://www.khanacademy.org
9	https://www.datacamp.com
10	https://www.programiz.com

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 TheoryTest	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

6. SEE – Practice Assessment Methodologies

S I . N O	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7. CIE Theory Test model question paper

Program		Information Science and Engineering			Semester -3rd	
Course Name		Data structure with python			Test	I/III
Course Code		25IS331	Duration	90 min	Marks	50
Name of the Course Coordinator:						
Note: Answer any one full question from each section. Each full question carries equal marks.						
Q.No	Questions			Cognitive Level	Course Outcome	Marks
Section - 1						

1	a)What is the difference between a tuple and a list in Python? Give a scenario where you would prefer using a tuple over a list. (5marks) b)Write a Python function that takes a list of integers as input and returns a new list where:(10 marks) • All even numbers are squared. • All odd numbers are multiplied by 3. Example Input: [1, 2, 3, 4, 5, 6] Example Output: [3, 4, 9, 16, 15, 36] c)Write a Python function that takes a list of tuples, where each tuple contains the name and the age of a person. The function should return the name of the oldest person.(10marks) Example Input: [('John', 45), ('Alice', 50), ('Bob', 30)] Example Output: 'Alice'			25
2	a)Describe the basic operations that can be performed on a set in Python. Provide examples of how to compute the intersection, union, and difference between two sets.(5marks) b)Write a Python function that takes a list of tuples, where each tuple contains the name and the age of a person. The function should return the name of the oldest person.(10 marks) Example Input: [('John', 45), ('Alice', 50), ('Bob', 30)] Example Output: 'Alice' c)Write a Python function that takes two sets and returns a tuple containing:(10 marks) • The intersection of the two sets. • The union of the two sets. • The difference between the first set and the second set. Example Input: set1 = {1, 2, 3, 4, 5} set2 = {4, 5, 6, 7} Example Output: ({4, 5}, {1, 2, 3, 4, 5, 6, 7}, {1, 2, 3})			
Section - 2				
3	Analyze the code and write the output <pre>def sum_nested_elements(nested_structure): total_sum = 0 stack = [nested_structure] # Initialize the stack with the outer structure print(stack) while stack: current = stack.pop() # Take the last element from the stack print(current) for item in current: if isinstance(item, (list, tuple)): # If it's a list or</pre>			25

	<pre> tuple, add to the stack print(item) stack.append(item) print(stack) elif isinstance (item, (int, float)): # If it's a number, add to the total total_sum += item return total_sum nested_list = [1, 2, (3, 4), [5, (6, 7)], 8] total = sum_nested_elements(nested_list) print(f"Total sum of elements: {total}") </pre>			
4	a) Create a Python program to Create a collection named trees – Oak, Pine, Mango, Neem, Maple, Teak, Mango, Neem, Maple, Pine, Mango, Oak, Pine, Mango, Peepal, Banyan, Sal, Compute and display the result with relevant messages for following: 1. Ask the user to input a tree name to count how many times it appears in the collection. 2. Ask the user to input a tree to find its position in the collection. 3. Get the last 5 tree names from the collection 4. Find the unique tree names from the collection Suggest the suitable python concept to solve above and justify your selection. b) Swami Vivekananda was a renowned Indian Hindu monk and one of the most influential spiritual leaders of the 19th century. For the above text, create a python program to 1. measure the length of the text. 2. count the number of words in the text. 3. Count the occurrence of these stop words – of, the, at, in, with 4. Capitalize these words and display the text – Narendranath, Vivekananda 5. Find numeric and alphanumeric words from the above text. Suggest the suitable python concept to solve above and justify your selection.			
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	
Course Name	Data structure with python			Test	
Course Code	25IS331	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
Problem Statement: Text and Numerical Data Separation with Word Classification Write a program that processes a given paragraph and performs the following tasks: 1. Read Input Data: ○ Read a text file containing text data. Each line of the file may contain multiple words. 2. Process Words: ○ Extract all the words from the file, ignoring special characters and punctuation. 3. Separate Text and Numerical Data: ○ Parse the paragraph to separate numerical data (e.g., numbers, digits) and text (e.g., words or characters). 4. Classify Words: ○ From the separated text, classify words into the following categories: ■ Uppercase Words: Words containing only uppercase letters (e.g., "HELLO"). ■ Lowercase Words: Words containing only lowercase letters (e.g., "world"). ■ Mixed Case Words: Words containing both uppercase and lowercase letters (e.g., "Python"). 5. Output: ○ The original paragraph. ○ A list of numerical data extracted from the paragraph. ○ Separate lists for Uppercase, Lowercase, and Mixed Case words.					50
Scheme of assessment (customize scheme as per requirement of your problem statement) a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10 b) Implementation and Execution - 30 c) Best Practices (Code Readability and Error Handling)- 10					
TotalMarks					50

Signature of the Course Coordinator

Signature of the HOD

9. SEE- Model Practice Question Paper

Program	Information Science and Engineering		Semester	3
Course Name	Data Structures using python	Course Code 25IS33I	Duration	180 min
Questions			CO	Marks
1. Design a student grade management system for your college to track and manage student grades. Let the system designed by you include the following requirements. • Add new student records with their name, subject, and grade. • Update a student's grade if there was a mistake. • Search for a student's grade by their name or subject. • Calculate and display the average grade for all students. The system should handle errors, such as trying to update or search for a non-existent student, and offer a simple menu for interaction.			1, 2, 3, 4, 5	50
Scheme of assessment (customize scheme as per requirement of your problem statement) a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10 b) Implementation and Execution - 30 c) Best Practices (Code Readability and Error Handling)- 10				
Total Marks				50

1) **Signature of the Examiner 1**

2) **Signature of the Examiner 2**

10. Suggestive practice programs:

The List is an Example and not inclusive of all possible activities of the course. Student and Faculty are encouraged to choose activities that are relevant to the topic

week	Suggestive practice programs
01	
02	1. Build a system to store email addresses of users subscribing to a newsletter. • No duplicate email addresses are allowed. • Add new email addresses. • Check if a specific email address is already subscribed. 2. Given two classes, store the register no. of students in each class using sets. Find students who are common to both classes. • Perform an intersection operation to find common students. • Perform a union operation to list all students from both classes. 3. Removing Duplicate Products Problem: Create a program to store product IDs of items in a warehouse. Use sets to ensure there are no duplicate product IDs. Tasks: • Add product IDs to the set. • Display all unique product IDs. 4. Storing Personal Information Problem: Create a program to store and display personal information of individuals. Each person's data

	should be stored in a tuple (Name, Age, City). Tasks: ● Access specific fields (e.g., print only the name of each individual). ● Try modifying an entry to demonstrate immutability.
03	1. Storing Favorite Books Problem: Write a program to keep track of your favorite books using a list. Tasks: ● Add books to the list. ● Remove a book if it's no longer a favorite. ● Display all favorite books. 2. Tracking Progress in a Game Problem: Store the levels completed in a game as a list. Tasks: ● Add a level when it's completed. ● Display the total number of levels completed. ● Access the last completed level using negative indexing. 3. Storing Product Prices Problem: Store the prices of products in a store using a dictionary. Use the product name as the key and the price as the value. Tasks: ● Add new products and their prices. ● Retrieve the price of a specific product. ● Update the price of a product. 4. Mapping Country Codes to Names Problem: Create a dictionary to map country codes (e.g., "US", "IN", "GB") to their respective country names. Tasks: ● Add new country codes and names. ● Retrieve the name of a country given its code.
04	1. Validating Email Addresses Problem: Write a program to check if an email address is valid. Tasks: ● Check if the email contains "@" and ".". ● Validate the length of the email address. ● Extract the username and domain name. 2. Counting Words in a Sentence Problem: Write a program to count the number of words in a sentence. Tasks: ● Split the sentence into words. ● Count and display the total number of words. 3. Writing a Daily Journal Problem: Create a program to maintain a daily journal. Each entry should be written to a file named <code>journal.txt</code>. Tasks: ● Take a journal entry as input from the user.

	● Append the entry to the file with a timestamp. ● Display the content of the journal. 4. Writing and Reading a Shopping List Problem: Create a program to store and retrieve a shopping list. Tasks: ● Write the shopping items to a file named <code>shopping_list.txt</code>. ● Read and display the list from the file. ● Add new items to the file.
05	1. Dividing Two Numbers Problem: Write a program to divide two numbers entered by the user. Tasks: ● Handle the <code>ZeroDivisionError</code> if the user attempts to divide by zero. ● Display an appropriate error message when an invalid input is entered (e.g., non-numeric values). 2. Validating User Input Problem: Write a program that asks for a user's age and checks if it is a valid number. Tasks: ● Handle invalid input errors (e.g., non-numeric values) using <code>ValueError</code>. ● Display an appropriate error message if the input is not a valid integer.
06	1. Factorial Calculation Problem: Write a program to calculate the factorial of a number using recursion. Tasks: ● Prompt the user to input a number. ● Calculate the factorial of the number using a recursive function. ● Display the result. 2. Fibonacci Sequence Problem: Create a program to generate the nth number in the Fibonacci sequence using recursion. Tasks: ● Prompt the user for the position (n) in the Fibonacci sequence. ● Use recursion to find the nth Fibonacci number. ● Display the result.
07	Library Management System (ADT and UDT) Problem: Create a library system where you can keep track of books. Implement a Book class (UDT) and use a List (ADT) to manage the collection. Tasks: ● Define a Book class with attributes like title, author, ISBN, and availability. ● Create a list of Book objects and allow adding, removing, and searching books by title or ISBN. ● Implement the option to check the availability of books and issue books to users.
08	Parentheses Balancing ● Problem Statement: Implement a program to check whether the parentheses in a mathematical expression are balanced using a stack. For example, the expression $(a + b) * (c + d)$ is balanced, while $((a + b) * (c + d)$ is not. ● Key Features: ○ Push each opening parenthesis onto the stack. ○ Pop when encountering a closing parenthesis, and check if it matches the

	opening parenthesis at the top of the stack. ○ If the stack is empty at the end of the expression, the parentheses are balanced.
09	Ticket Reservation System ● Problem Statement: Create a ticket reservation system for a bus service, where passengers are served in the order of their booking. New bookings are added to the queue, and they are processed in the same order. ● Key Features: ○ Enqueue: Add new ticket bookings to the queue. ○ Dequeue: Process bookings in the order they were made. ○ Peek: View the next booking to be processed.
10	Implement a Shopping Cart ● Problem Statement: Build a simple shopping cart system where each item has a product name and price. The cart should allow adding new items, removing items, and displaying the current contents of the cart. ● Key Features: ○ Add new items (product name, price) to the cart (insert at the end). ○ Remove an item by product name (delete by value). ○ Display all items in the cart (traverse the list).
11	Train Carriage System ● Problem Statement: Implement a train carriage system where each node represents a train carriage, and the train can move forward or backward by attaching or detaching carriages. ● Key Features: ○ Add carriages to the front or back of the train. ○ Remove carriages from the front or back. ○ Move through the carriages (using both forward and backward traversal)
12	Family Tree Representation ● Problem Statement: Create a family tree where each person has a name and a list of children (if any). The root of the tree represents the oldest ancestor, and each node represents a person. ● Key Features: ○ Add family members and their relationships (parent-child). ○ Traverse the tree to display the family history (e.g., ancestors or descendants). ○ Find all children of a specific person. ○ Search for a person in the family tree.
13	Family Tree Representation (Pre-order/Post-order Traversal) ● Problem Statement: Implement a family tree where each node represents a family member, and each member can have multiple children. Use pre-order and post-order traversal to display family members. ● Key Features: ○ Add family members and their relationships (parent-child). ○ Use pre-order traversal to display the root family member followed by their descendants. ○ Use post-order traversal to display descendants first, followed by the ancestor.

11. Rubrics for Assessment of Activity(QualitativeAssessment)

Sl. No.	Dimension	Excellent (10)	Good (8)	Satisfactory (6)	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/ Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
	TOTAL					40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	i3 or i5	30
02	Internet Connection	100 Mbps	



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	3
Course Name	System and Network Administration	Type of Course	Integrated
Course Code	25IS34I	Contact Hours	7 Hours per week Total 91
Teaching Scheme	3:0:4	Credits	5
CIE Marks	50	SEE Marks	50 (Practice)

1. Rationale:

The course System and Network Administration is designed to equip learners with the skills and knowledge required to manage, maintain, and troubleshoot computer systems and networks. By combining theoretical knowledge with hands-on practice, this course prepares learners for a critical role in the IT infrastructure of organizations. As technology forms the backbone of modern organizations, proficient administrators are critical to ensuring systems remain secure, reliable, and efficient.

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

CO-01	Identify hardware specifications & assemble a computer as per given technical specifications.
CO-02	Install an Operating system, configure and modify administration settings, carry out basic troubleshooting & performance monitoring
CO-03	Comprehend the file system, accomplish Formatting & Partitioning a storage drive, back up data and use file systems' recovery tools effectively.
CO-04	Configure Network, check and update network administration settings, provide basic network security & sharing activities.
CO-05	Plan Layout for setting up Lab & follow e-waste protocols while disposing computer hardware, to ensure compliance with all required state pollution control board regulations.

3. Course Content

Week	CO	PO	Lecture(3HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1	Computer hardware categories & their usage. External components Input devices Output devices Internal components Processing devices Storage devices Network devices Cables and connectors used with computers	Hardware Identification: 1. Computer Case- Types, Features 2. Identify the front and rear panel controls and ports on a PC cabinet. 3. A look inside the computer case. 4. Identify different cables and CMOS battery, front & back panel connectors, power connectors. 5. Identify Storage devices(Hard disk, SSD), PSU, Cooling systems & their connection to the motherboard. 6. Identify the networking devices connected to the computer.
2	1	1,5	Power supply unit - SMPS What is SMPS? Need for SMPS SMPS parts and connectors. Working principles of SMPS Types of SMPS SMPS voltage color codes Measuring SMPS voltage levels. How to choose right power supply for PC Common issues in SMPS. Troubleshooting common problems in SMPS	1. Unmount the power supply from the PC cabinet. Identify the types of output connectors. 2. Identify output voltages using color coding. 3. Measure voltage levels using a multimeter. 4. Mount the power supply into the PC cabinet, connect different components and test the PC. 5. Troubleshoot Power supply through SMPS fan. 6. Diagnose power supply faults using PSU Tester.
3	1	1,5	Exploring Motherboard What is a motherboard? Types of motherboard(Form factor) Components of Motherboard How does a motherboard work? Main functions of motherboard Popular Motherboard Manufacturers. Desktop motherboard Vs Laptop motherboard	1. Identify various components of the motherboard.(Draw motherboard layout) 2. Compare motherboards of various types 3. Precautions to be taken before removing the motherboard from the PC cabinet. 4. Using the CPUID CPU-Z tool, find different features of the CPU. 5. Check the Electric flow path and data flow path

			Common motherboard issues and maintenance. Mother troubleshooting techniques Usage of CPUID CPU-Z tool.	6. Using Windows resource monitor to troubleshoot performance issues. 7. Using the CPUID CPU-Z tool, identify the CPU cache features of your working system.
4	1	1,4,5	Chipset Architecture and components Northbridge Southbridge Booting process (FIRMWARE) • POST: • BIOS/UEFI • Bootloader Execution • OS Kernel Loading • System Processes Initialization • User Authentication	1. Hands-on identification of Northbridge and Southbridge chips on motherboards 2. Identify the components connected to the north bridge and south bridge. 3. Implement basic power management features using BIOS/UEFI settings. 4. Checking System Information in BIOS/UEFI & Change the Boot Order 5. Analyze the chipset design of a specific device (e.g., smartphone, laptop) and suggest improvements
5	2	1,2,5	Operating System Installation Creating OS Image (Bootable Device) Operating system : Functions and types Understanding file systems and partitions	1. Install and boot Windows 10/11 or any Linux OS 2. Modify basic admin settings like: a. Changing Desktop Background b. Inverting and rearranging screen c. Testing Microphone 3. Document OS installation procedures Windows Linux
6	2	1,2,4,5	Windows Command line interface Basic Navigation commands File and directory management commands writing simple PowerShell scripts. File security. User Account and Permission Management User Profile Issues: Create a	1. Using command prompt or powershell, Execute basic navigation commands, Execute file and directory management commands. 2. Use command line to encrypt and decrypt files and folders 3. Adding, modifying, and deleting user accounts. 4. Windows: Using User Accounts in Control Panel or via PowerShell (New-LocalUser,

			New Profile, Profile Repair	Set-LocalGroup). 5. Create a New Profile, Profile Repair 6. Scripts in powershell like read/write data from console/file
7	2	1,2,4,5	OS configuration and Administration System Startup Issues and solutions: Safe Mode, Automatic Repair, System Restore, system reset, Image recovery Performance Issues: Task Manager, Resource Monitor, Disk Cleanup, Check for Malware, Update Drivers, Check Disk Health	1. Solve system startup issues using Safe Mode, Automatic Repair, System Restore, system reset, Image recovery 2. Check and/or Solve Performance Issues using Task Manager, Resource Monitor, Event Viewer, Disk Cleanup, Check for Malware, Update Drivers, Check Disk Health 3. Document about “Blue Screen of Death (BSOD), or bluescreen error” 4. Document about “white screen error”
8	3	1,2,4,5	File system overview, file systems, types, properties, conversion from one file type to another (ex:.doc/docx to pdf) Configuring system: Disk partitioning, Shrink, extend. Data recovery: Recover files, recover apps and the registry; Recover Windows 10/11. Configure and manage windows updates: auto/manually; testing and troubleshooting updates.	1. Practice data recovery methods 2. Working with a task manager to troubleshoot configuration and other performance related issues. (Startup tasks and auto services) using comand line and GUI 3. Working with a task scheduler. 4. Customizing windows desktop 5. System restore Setting up file permissions (NTFS in Windows)
9	3	1,2,4,5	Monitor and manage: Performance monitoring; optimize windows services; tune scheduled tasks. Customizing windows desktop. File and System Corruption: - System File Checker (SFC) - Deployment Imaging	1. Monitor the processes, System performance using Task manager (Windows)/ Process related Commands in Linux. 2. Troubleshoot the system delaying for booting (disabling unnecessary Startup applications, Services) 3. Work with task scheduler:

			Service and Management Tool (DISM) • Check Disk Utility Drivers Management: Device drivers, Network drivers, System drivers	Windows/Linux 4. Install printer driver, 5. Network driver disable/ enable, 6. System driver update
10	4	1,2,3,4,5	Network Configuration and Management : IP Addressing (Static IP Addressing: Dynamic IP), Subnetting, DNS Configuration, Gateway Configuration,VLAN Configuration Network Troubleshooting: Identify the Problem, Establish a Baseline,Check Physical Layer,Verify the Data Link Layer,Test Network Connectivity,Use Traceroute,Check IP Address Configuration,DNS Resolution Issues,Examine Firewall and Security Settings Network Security: Firewall, Network profile, Managing wireless network Network Sharing	1. Configure Network Connections, Configure IP Settings, Set DNS Server, Network Adapter Information, Disable or Enable Network Adapter. 2. Ping, Trace Route, Checking Open Ports, Network Statistics, Flush DNS Cache, Reset TCP/IP Stack, 3. Windows Firewall Configuration, Configure Network Profiles, Manage Wireless Networks 4. Share any Files and Folders with other System in the network. 5. Access another computer using remote desktop app
11	4	1,2,4,5	Network Monitoring and Reporting: Monitor Network Activity, View Network Usage Advanced Network Configuration: Create and Manage Virtual Networks (Hyper-V Manager), Manage Routing (Modem) and Remote Access Virtual Private Network: Definition, Types of VPN, Implementation	1. Monitor network activity using Task manager and resource monitor 2. Install a VM(Hypervisor) document VM Network Utilization Metrics 3. Run basic network commands in guest OS installed on VM 4. Download and install any third party VPN app and connect internet through VPN
12	5	7,8	E-waste management:	• Visit https://greene.gov.in and

			E-Waste: Definition and types of electronic waste (e.g., computers, smartphones, home appliances). Global statistics on e-waste generation. Environmental and health impacts of improper e-waste management. E-Waste Management Lifecycle: Lifecycle of electronic devices: production, usage, and disposal. Processes in e-waste recycling: collection, sorting, and recovery. Policies and Regulations on E-waste: Global e-waste policies National laws and policies on e-waste	https://kspcb.karnataka.gov.in/ to find the latest regulations and policies taken up by the Government of India. - Visit a nearby e-waste management plant and understand the management process. - Document how to reduce carbon footprint
13	1,2,3,4,5	1,2,3,4,5,6	Mini Project Activity- Layout Management Lab setup Management: Resource purchase, Installation, Electrical safety measurement	Design and document a layout and Setup a computer lab in your Polytechnic along with safety measures and all necessary applications and network security measures.

4. Learning resources:

Week	References
01	https://discover.hubpages.com/technology/The-Four-Main-Categories-Of-Computer-Hardware-Parts https://www.javatpoint.com/what-are-the-computer-cables
02	https://www.intel.com/content/www/us/en/gaming/resources/power-supply.html https://en.wikipedia.org/wiki/Switched-mode_power_supply https://www.amd.com/en/resources/support-articles/faqs/GPU-104.html
03	https://www.geeksforgeeks.org/what-is-a-motherboard/ https://www.studocu.com/en-us/document/purdue-university-global/information-technology-concepts/12-exploring-motherboards-processors-and-memory/37160479
04	https://www.baeldung.com/cs/chipset https://www.javatpoint.com/booting-in-operating-system https://learn.microsoft.com/en-us/windows-hardware/drivers/bringup/boot-and-uefi
05	https://www.javatpoint.com/how-to-install-windows-10 https://answers.microsoft.com/en-us/windows/forum/all/steps-to-install-windows-10/a499989c-f31f-49d0-8dbb-3b6425a5f47b https://www.geeksforgeeks.org/how-to-install-ubuntu-desktop/
06	https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands https://www.freecodecamp.org/news/command-line-commands-cli-tutorial/

	https://www.geeksforgeeks.org/most-useful-cmd-commands-in-windows/
	https://devblogs.microsoft.com/scripting/table-of-basic-powershell-commands/
07	https://learn.microsoft.com/en-us/troubleshoot/windows-client/performance/windows-boot-issues-troubleshooting https://learn.microsoft.com/en-us/troubleshoot/windows-client/performance/windows-startup-issues-troubleshooting https://support.microsoft.com/en-us/windows/system-configuration-tools-in-windows-f8a49657-b038-43b8-82d3-28bea0c5666b
08	https://learn.microsoft.com/en-us/windows-server/storage/disk-management/overview-of-disk-management https://support.microsoft.com/en-us/windows/windows-file-recovery-61f5b28a-f5b8-3cc2-0f8e-a63cb4e1d4c4
09	https://support.microsoft.com/en-us/windows/system-configuration-tools-in-windows-f8a49657-b038-43b8-82d3-28bea0c5666b https://support.microsoft.com/en-us/windows/update-drivers-manually-in-windows-ec62f46c-ff14-c91d-eead-d7126dc1f7b6
10	https://learn.microsoft.com/en-us/training/paths/configure-networking-windows-clients/ https://support.microsoft.com/en-us/windows/change-tcp-ip-settings-bd0a07af-15f5-cd6a-363f-ca2b6f391ace https://support.microsoft.com/en-us/windows/setting-up-a-wireless-network-in-windows-97914e31-3aa4-406d-cef6-f1629e2c3721 https://learn.microsoft.com/en-us/windows/security/operating-system-security/network-security/windows-firewall/ https://www.windowscentral.com/8-networking-command-tools-every-user-should-know-windows-10
11	https://www.vmware.com/topics/virtual-networking https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn
12	https://greene.gov.in/ https://kspcb.karnataka.gov.in/

5. CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1 Theory Test	4	90	50	
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 Theory Test	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

6. SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

7 CIE Theory Test model question paper

Program		Information Science and Engineering			Semester -III	
Course Name		System and Network Administration			Test	I
Course Code		25IS34I	Duration	90 min	Marks	50
Name of the Course Coordinator:						
Note: Answer any one full question from each section. Each full question carries equal marks.						
Q.No	Questions			Cognitive Level	Course Outcome	Marks
Section - 1						
1	a) Categorize the given devices/components into multiple categories i. Mouse ii. Phone Display iii. RAM iv. RJ45 v. Flash Drive vi. NIC vii. Printer viii. Projector ix. Cache -05			L3	1	25
	b) Ajay a computer science diploma II year student. One of his friends wanted to buy a new desktop, so he asked Ajay’s suggestions, feeling he might know more about computers. What are your suggestions to him with respect to hardware? -10			L3	1	
	c) Compare the hardware components of a computer with your mobile phone. -10			L2	1	
2	a) How can you diagnose power supply faults using PSU Tester? 5			L3	1	
	b) Rachana was sitting on a plastic chair before trying to check the system unit cabinet. Though the power supply is not connected she got an electric shock. How can you analyze this problem? What are the safety measures for this? (static electricity) -10			L3	1	
	c)How does SMPS work? -10			L2	1	
Section – 2						
3	a) Anil and Hruthvik are debating whether the north bridge is faster or the south bridge is faster? How can you conclude it? Justify your answer. -05			L3 L3	1 1	25
	b) One of the computers in your lab was hanged when Vedant					

	ran application 'X'. He doesn't want to restart the computer but has to kill application 'X'. How can you help him? -10 c) Khushi wants to install an operating system from his pen drive. But it is not loading. Someone told her to change it to BIOS. She does not understand it clearly. Can you help her with this regard 10	L3	1,2	
4	a) How can you differentiate single boot virtual machine installation? -5 b) Compare windows/Linux OS with android OS -10 c) Rehman installed an older version of the operating system on his computer. Some of his hardwares like Wi-Fi, is not working. How can you resolve this issue? (Chipset not installed) -10	L2 L2 L3	1,2 1,2 1,2	
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator Signature of the HOD Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	1
Course Name	System and Network Administration			Test	II
Course Code	25IS34I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
1. You are instructed to install newer version of Operating system in a computer in your lab. Decide the OS, document hardware requirement for the same, before installing the OS check the BIOS/UEFI settings and do necessary updates in it. Install OS on right PC. Create a new account name CSE_LAB1_01 with a strong password				1,2,3	50
Scheme of assessment					
a) Choosing OS and document Hardware					10
b) BIOS/UEFI settings					10
b) Installation of OS and related operations					15
c) OS related operations					10
d) Portfolio Evaluation					05
Total Marks					50

Sign of the Course Coordinator

Signature of the HOD

9. SEE-Model Practical Question Paper

9. SEE Model Practical Question Paper					
Program	Information Science and Engineering		Semester		
Course Name	System and Network Administration		Marks	50	
Course Code	25IS34I		Duration	180 Min	
Questions			Cognitive Levels	Course Outcomes	Marks
Section -1	You are tasked with setting up and maintaining a computer system for a small office environment. The goal is to ensure that the hardware components are correctly identified, software is installed and configured, issues are resolved efficiently, and the network is operational for seamless communication. Requirements: Hardware: Verify hardware specifications and compatibility with the system requirements. Software Installation: Install an operating system (e.g., Windows/ Linux) on a computer. Install essential drivers and utility software required for system functionality Diagnose ,resolve and document any hardware and software issues faced Network Configuration: Set up a basic wired or wireless network for the office, ensuring connectivity between devices		L3	1,2,3,4,5	45
Section -2	Portfolio Evaluation/Viva		L3		5

2) Signature of the Examiner 1

2) Signature of the Examiner 2

10. Rubrics for Assessment of Activity (Qualitative Assessment)

Sl. No.	Dimension	Excellent	Good	Satisfactory	Needs improvement	Score
		9-10	7-8	5-6	Below 5	
1	Knowledge of Hardware Components	Demonstrates an in-depth understanding of all hardware components, their roles, and interconnections with accurate terminology.	Understands most hardware components and their roles, with minor gaps or inaccuracies.	Basic understanding of hardware components but struggles with details or technical terminology.	Limited understanding of hardware components and their functions.	8
2	Configuration and Installation Skills	Configures and installs hardware/software accurately without assistance, following industry best practices.	Configures and installs hardware/software correctly with minimal guidance; few errors.	Completes configuration and installation with frequent guidance; some errors in execution.	Unable to complete configuration or installation without significant errors or assistance.	6

3	Troubleshooting and Problem-Solving	Identifies and resolves issues systematically and efficiently using appropriate tools and methods.	Diagnoses and resolves most issues with minor inefficiencies or errors	Can diagnose and resolve basic issues but struggles with more complex problems.	Unable to diagnose or resolve issues without significant assistance.	2
4	Documentation and Reporting	Produces detailed, well-organized, and professional documentation with no errors.	Documentation is clear and organized but contains minor errors or omissions.	Documentation is basic, with several errors or incomplete details.	Documentation is incomplete, unclear, or missing important details.	3
5	Teamwork and Collaboration	Actively contributes to team efforts, communicates effectively, and assumes leadership when necessary.	Contributes effectively to the team with good communication; may rely on others for leadership.	Participates in team activities but lacks initiative or effective communication.	Limited participation or ineffective collaboration in team tasks.	2
Total/50						21/50

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers		10
02	PSU Tester		10
03	Multimeter		10
04	Individual components- SMPS/PSU	400 watts	10
05	Individual components- SMPS/PSU	800 watts	10
06	Motherboard	ATX	10
07	Motherboard	Micro ITX 10	10
08	RAM stick	DDR3	10
09	RAM stick	DDR4	10
10	CMOS battery		10
11	Windows 10/11 OS user license for multi users		-
12	POST diagnostic card		10

SEMESTER 4



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	4
Course Name	Operating System & Administration	Type of Course	Integrated
Course Code	25CS41I	Contact Hours	8 per week
Teaching Scheme	4:0:4	Credits	6
CIE Marks	50	SEE Marks	50 (Theory)

1. Rationale:

This course provides a comprehensive introduction to the fundamental concepts of operating systems, Linux operating system and shell programming including their structure, functionality, and design principles. Students will gain hands-on experience with process management, memory management, file systems, and learn how to navigate and utilize Linux environments, and write scripts to automate tasks and manage system resources effectively.

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

CO-01	Understand and analyze basic concepts of Operating system and memory management technique and virtualization
CO-02	Analyze and apply process management and scheduling techniques and understand file systems.
CO-03	Explore and communicate technical concepts related to storage, networking, and security
CO-04	Perform basic system administration tasks like monitoring system performance, managing users, and handling permissions using shell scripting.
CO-05	Analyze the power and energy-efficient optimization strategies in OS and Implement cloud-based solutions for advanced OS technologies.

3. Course Content

Week	CO	PO	Theory	Practice
1	1		Introduction to Operating Systems- - Overview of Operating System - Need for OS, Structure - OS Types (Batch, Multitasking, Real-time, Distributed, etc.) - Types of OS installation - Dual mode operation - Kernel and microkernel - Functions of OS (resource management, process management, memory management, etc.) - Booting process and system initialization	1. Demonstrate the installation of Windows & Linux OS 2. Demonstrate Boot methods using videos 3. Demonstrative videos on File System and formatting 4. Explore the post installation tasks.
2			Memory management- Memory Organization: RAM, Cache	1 Install and configure configuring virtual machines (VMs) 2 Download and install a terminal

	1,2		Memory, Virtual Memory - Memory Allocation: Contiguous Memory Allocation, Paging and Page Tables, Segmentation Virtualization technology- - Definition and importance of virtualization - Virtual Memory: Demand Paging, Page Replacement Algorithms (FIFO, LRU, Optimal, and Clock algorithms)	emulator 3 Demonstration through videos. 4 Commands to view memory consumption (free, free -b, free -k, free -g)
3	2		File system – - Definition and role of file systems in operating systems - Types of file systems (local vs. distributed) - File system security issues - Access control lists (ACLs) vs. File permissions - Next-generation file systems (e.g., Fusion File System, F2FS for SSDs) - The role of AI in file system management	File and Directory commands: - Create and delete directories and files, File movement, copy commands, Pipes - Commands for viewing File, File comparison, File manipulation, Altering file permission, File compression and decompression. - Text processing commands (grep, sort, uniq, cat, cut, paste, head, tail, wc, tr).
4	3		Process Management – - Definition and components of a process - Process states and state transitions (new, ready, running, waiting, terminated) - Process control block (PCB) - Process scheduling (long-term, short-term, and medium-term scheduling) - Inter-process communication (IPC): shared memory, message passing	1 process creation and management commands - system calls fork() and exec(); bg, fg, pkill, nice, top, ps.
5	3		Process Synchronization and Concurrency- - Critical Section Problem Synchronization Mechanisms: - Semaphores: Binary and counting. - Monitors and Condition Variables. Deadlock: - Necessary Conditions for Deadlock. - Deadlock Prevention and Avoidance Strategies. - Resource Allocation Graphs. - Deadlock Detection and Recovery	- Demonstration through videos. - Commands to exhibit thread concepts. (top, htop, and ps to monitor system threads). - Design a system to prevent deadlock in a multi-processing environment using resource allocation graphs or safe states.

6	3,4	I/O Management and Disk Scheduling- - I/O Devices and Device Drivers: Types of I/O devices: Hard Disk, Keyboard, Display, Printers. Role of Device Drivers in OS. - Types of Buffers: Single, Double, Circular. - Importance of Buffering in I/O Operations, - Direct Memory Access (DMA): Overview of DMA. - Use of DMA in I/O operations.	1 Monitoring I/O on a System: command in Linux is used to monitor CPU and I/O statistics(iostat, iotop, dstat) 2. Discuss the impact of fragmentation on both read and write operations and how the operating system can mitigate fragmentation. 3.
7	3	Storage management: - Disk partition, formatting, mounting; - Logical Volume Management (LVM)- Use of LVM, creating Volume groups, logical volume and disk mirroring, - Extend Disk using LVM, Adding Swap Space - Introduction to RAID – Hardware & Software, RAID levels	1. Basic commands for storage partitions. 2. Install and configure LVM. 3. Simulate RAID configurations and understand how data is stored and recovered.
8	3	Networking, security and protection in Operating System: Networking Basics: - Sockets and networking in OS - Remote Procedure Calls (RPC) - Distributed File Systems (DFS) - Security Principles - Authentication, Authorization, and Access Control - Encryption and Data Integrity - User Authentication - Passwords, multi-factor authentication	1. Work on user accounts: useradd, passwd, userdel, usermod,groupadd, groupmod,gpasswd, groupdel; system-config.
9	4	Linux administration - History and Overview of Linux - Basic Linux Commands (ls, cd, cp, mv, rm, mkdir, rmdir) - Permissions and Ownership - File Types and Attributes - Working with Files and Directories (find, locate, grep, tar, zip/unzip) - Text Processing Utilities (cat, more, less, head, tail, grep etc)	1.Use mkdir to create a new directory named OS_lab. 2.Navigate into the OS_lab directory using cd. 3.Create subdirectories named data, scripts, and logs 4. Demonstrate grep 5. use text processing cmd cat, more, head.
10		Shell Programming	1.Use variables and take user

	4	• Shell Overview and Types (Bash, Zsh, Ksh, Tcsh) • Role of the Shell in Unix-like OS • Basic Shell Commands (ls, cd, pwd, cat, echo, etc.) • Shebang (! /bin/bash) in scripts • Writing Simple Shell Scripts • Creating variables, accessing variables using \$ and \${}, Special variables (\$1, \$2, \$#, \$@, \$?) • Input/Output, and Control Structures	input in a script. 2. Write a shell script that prints "Shell Scripting is Fun!" on the screen 3. Write a script to print a given number in reverse order.
11	4	Advanced Shell Scripting • Functions and Error Handling • Arrays and String Manipulation • Script Debugging Techniques • Advanced Topics: Process Substitution, Command Substitution	1. Write a basic shell script to automate a task. 2. Write a shell script to check to see if the file "file_path" exists. If it exist, display "file_path passwords are enabled." Next, check to see if you can write to the file. If you can, display "You have permissions to edit "file_path."" If you cannot, display "You do NOT have permissions to edit "file_path"" 3. Write a shell script that consists of a function that displays the number of files in the present working directory. Name this function "file_count" and call it in your script.
12	5	Performance and Optimization Performance Evaluation System performance metrics: CPU utilization, throughput, latency System Tuning Kernel tuning, memory tuning Multicore and Multithreading Parallel computing models and threading Energy-Efficient Operating Systems Power management and optimization in OS 2.	1. Consider an application that simulates a large number of particles interacting with each other in a physics-based simulation. How would you use multicore processing and multithreading to speed up the simulation? Outline the steps to divide the work into parallel tasks and ensure proper synchronization. 2. Using a parallel computing model like the SPMD (Single Program Multiple Data) model, describe how you would break down the problem and distribute the work across multiple threads or processors. How would you

				ensure that data dependencies are handled correctly and that the workload is balanced?
13	5		- Real-time operating systems (RTOS) and their features - Kernel development and module programming - OS-level debugging and profiling tools - Cloud-based operating systems (Google Fuchsia, Chrome OS) - Emerging OS technologies (Microservices architecture, Containers)	- implement a simple real-time system and understand key concepts of RTOS such as tasks, scheduling, and interrupts. - understand the concept of cloud-based operating systems, their architecture, and the benefits of using cloud OS in real-world applications.

4. Learning Resources

Sl No	References
1	Operating System Concepts by Silberschatz, Galvin, and Korth
2	Supplementary Text: Modern Operating Systems by Andrew S. Tanenbaum
3	The Linux Command Line: A Complete Introduction by William E. Shotts Jr.
4	Advanced Programming in the UNIX Environment by W. Richard Stevens, Stephen A. Rago
5	Supplementary Text: Bash Cookbook by Carl Albing, JP Vossen
6	https://methodist.edu.in/web/uploads/labmanual/OS%20Lab%20Manual.pdf
7	https://it.mitindia.edu/PDF/manuals/IT7411.pdf
8	https://people.iitism.ac.in/~download/lab%20manuals/cse/CSC211.pdf

5. CIE Assessment Methodologies

CIE Assessment Methodologies					
Sl . No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

6. SEE - Theory Assessment Methodologies

Sl.No	SEE - Theory Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Theory	90	50	20

7. CIE Theory Test model question paper

Program		Information Science and Engineering			Semester -4	
Course Name		Operating System & Administration			Test	
Course Code		25CS41I	Duration	90 min	Marks	50
Name of the Course Coordinator:						
Note: Answer any one full question from each section. Each full question carries equal marks.						
Q.No	Questions			Cognitive Level	CO	Marks
Section - 1						
1	a) What are the major components of an operating system			L2	1	25
	b) How does an operating system manage hardware resources like CPU, memory, and I/O devices			L3	1	
	c) What is a virtual machine, and how does an OS support virtualization?			L2	2	
2	a) Why is it necessary for an operating system to have both User Mode and Kernel Mode			L2	1	
	b) What is a system call, and how does it interact with the OS			L2	1	
	c) How does a Command-Line Interface (CLI) differ from a Graphical User Interface (GUI) in terms of functionality and user experience?			L3	2	
Section - 2						
3	a) In what scenarios would you choose Shortest Job First (SJF) over First-Come, First-Served (FCFS)			L2		25
	b) What factors influence the time taken during a context switch, and how can this time be minimized?			L2		
	c) Consider the following processes with their arrival times and burst times using FCFS Problem.			L3		
	i) Calculate the completion time, turn-around time, and waiting time for each process					
	ii) Find the average turn-around time and waiting time for the processes.					
	Process	Arrival Time	Burst Time			

	P1 0 8 P2 1 4 P3 2 9 P4 3 5			
4	a) Explain any algorithms used for detecting deadlocks. b) Explain the two main types of threads supported by most operating systems c) Using the Round Robin scheduling algorithm, determine the order in which the processes will be executed. Calculate the average turnaround time and waiting time for the processes. ProcessBurst Time P110 P25 P38 P46	L2 L2 L3		
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	4
Course Name	Operating System & Administration			Test	
Course Code	25CS41I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
1) Write a shell script to generate a multiplication table of the given number.					
2) Write a shell script to display files which has read, write and execute Permissions				4	10
3.) Execute commands - system calls fork() and exec(); bg, fg, pkill, nice, top, ps.				4	10
4)We need to synchronize files between two directories: /source_dir and /backup_dir. The synchronization should ensure that:				2	10
- Any new or modified files from /source_dir are copied to /backup_dir. - Deleted files from /source_dir should also be removed from /backup_dir. - The script should be able to run automatically at regular intervals (e.g., every hour). - Any issues (e.g., files that couldn't be copied) should be logged.				4	20
Total Marks					50

Signature of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

9. SEE–Model Theory Question Paper

Program	Information Science and Engineering	Semester	4
Course Name	Operating System & Administration	Marks	50
Course Code	25CS41I	Duration	90 Min

Note: Answer any one full question from each section. Each full question carries equal marks.

Q No	Questions	Cognitive Levels	Course Outcomes	Marks
Section -1				
1	a) Explain different types of Operating system and List its functions.	L2	1	5
	b) What is the difference between user space and kernel space in an OS structure?	L2		5
	c) Explain how a virtual machine monitor (VMM) interacts with the kernel and the user space of an operating system.	L3		10
2	a) Describe the purpose of a distributed operating system and how it differs from a centralized operating system.	L2	1	5
	b) How does a Command Line Interface (CLI) differ from a Graphical User Interface (GUI)?	L2		5
	c) Why is Dual Mode Operation essential in modern multi-tasking operating systems like Windows or Linux?	L3		10
Section -2				
3	a) What is the role of the program counter in the process Control Block?	L2	2	5
	b) Consider the following processes with their burst times and a time quantum of 4:	L3		10
	i) Using the Round Robin scheduling algorithm, determine the order in which the processes will be executed.			
	ii) Calculate the average turnaround time and waiting time for the processes.			
	Process Burst Time			
P1	10			
P2	6			
P3	4			
P4	1			
	c) What is the relationship between threads and processes in a multi-threaded system	L2		5

4	a) What are the differences between deadlock prevention, deadlock avoidance , and deadlock detection strategies.	L2		5
	b) In the producer-consumer problem , how can semaphores be used to synchronize the producer and consumer processes?	L3	2	10
	c) Explain the state transition diagram of Process.	L2		5
Section -3				
5	a) Consider a system with 3 frames in physical memory and the following reference string: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 4 Apply the First-In-First-Out (FIFO) page replacement algorithm and find: i)The number of page faults ii)LRU	L3	3	10
	b) How does paging differ from segmentation in memory management	L2		5
	c) Explain the basic structure of a file system	L2		5
6	a) How do you compare the performance of different disk scheduling algorithms	L3	3	10
	b) What is a directory structure , and how are files organized within it	L2		5
	c) What is the purpose of buffering in I/O operations?	L2		5
Section -4				
7	a) After adding a new CPU-intensive application, your system performance has dropped. What steps would you take to assess whether the CPU is a bottleneck?	L3	4	10
	b) Your laptop runs out of battery quickly, even when you're not using many applications. How would you optimize power usage to extend battery life?	L3		10
8	a) How do you handle error handling in shell scripts	L3	4	10
	b) Write a shell script to count the number of lines, words, and characters in a file.	L3		x 10
Section -5				
9	a) What is a Real-Time Operating System (RTOS), and how does it ensure that tasks are completed within strict timing constraints?	L3	5	10
	b) How can a cloud-native OS improve scalability and resource management in a cloud environment?	L3		10
10	a) How does an OS-level debugger interact with system processes to troubleshoot issue.	L3	5	10
	b) Explain how perf can be used to profile CPU performance, and how it helps in identifying bottlenecks in system performance	L3		10

10. Suggestive Activities for Tutorials:

The List is an Example and not inclusive of all possible activities of the course. Student and Faculty are encouraged to choose activities that are relevant to the topic

Sl.No.	Suggestive Activities for Tutorials
01	Implement simple system calls like read(), write(), open(), and close(), representing interaction between a user program and the kernel space.
02	Simulate and calculate various performance metrics such as turnaround time , waiting time , and throughput for each algorithm.
03	Simulate a deadlock situation in a multithreaded environment (e.g., using resource allocation graphs or the Banker's algorithm).
04	Simulate segmentation and paging techniques and manage memory for processes.
05	Implement a file allocation strategy such as contiguous allocation , linked allocation , or indexed allocation .
06	Identify critical systems and define the procedures to recover them in the event of a natural disaster, cyberattack, or hardware failure.

11. Rubrics for Assessment of Activity(Qualitative Assessment)

Sl. No	Dimension	Beginner	Intermediate	Good	Advanced	Expert	Students Score
		2	4	6	8	10	
1	Understanding of Key Concepts	Lack of understanding of basic concepts	Lack of understanding basic concepts, with multiple misconceptions.	Demonstrates basic understanding with some inaccuracies	Shows good understanding with minor errors or missing details.	Demonstrates comprehensive understanding of OS concepts	8
2	Problem Solving	Unable to solve problems without significant external help.	Requires significant assistance or struggles to resolve issues.	Solves problems independently with minimal guidance	Solves most problems independently, though may need assistance for complex issues.	Effectively solves any problems that arise during the installation/setup process with minimal guidance.	6
3	Depth of Analysis	Lack of providing analysis with some examples	Provides little analysis or examples.	Provides limited analysis with few examples or applications	Provides adequate analysis with some examples or applications	Provides deep insights with correct and relevant examples.	2
4	Documentation and Reporting	Provides little or no documentation, or it is very unclear.	Provides an incomplete or unclear report, lacking sufficient detail.	Provides a mostly clear report with some missing details.	Provides a clear report with some details or unclear sections.	Provides a clear, detailed report or documentation on the setup process, troubleshooting steps, and results.	2

	Average Marks=(8+6+2+2)/4=4.5	5
--	-------------------------------	---

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	i3 or i5	30
02	Internet Connection	100 Mbps	



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	4
Course Name	Object Oriented Programming with Java	Type of Course	Integrated
Course Code	25CS42I	Contact Hours	8 Hours / Week
Teaching Scheme	L: T:P ::4:0:4	Credits	6
CIE Marks	50	SEE Marks	50 (Theory)

1. Rationale

This course is designed to equip students with a comprehensive understanding of object-oriented programming (OOP), a foundational paradigm in modern software development. The focus is on enabling students to write maintainable, modular, and scalable code while gaining practical experience in implementing these concepts using Java. As a versatile programming language, Java is widely used for enterprise applications, mobile apps, and web services, making it an ideal choice for mastering OOP concepts and SOLID principles.

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

CO-01	Explain the core principles of object-oriented programming and their significance in software development.
CO-02	Create Java programs that leverage object-oriented concepts to solve real-world problems effectively.
CO-03	Refactor code to enhance readability, maintainability, and efficiency while adhering to SOLID principles.
CO-04	Demonstrate the use of exception handling mechanisms in Java to identify and manage runtime errors, ensuring application reliability.
CO-05	Establish database connectivity using Java and perform data operations

3. Course Content

Week	CO	PO	Theory	Practice
1	1	1	Introduction to Object-Oriented Programming Define Object-Oriented Programming (OOP) and its key objectives. Principles of OOP: <i>Encapsulation, Abstraction, Inheritance and Polymorphism</i> Real-world Examples of OOP Overview of SOLID Principles Importance of SOLID Principles Overview of Java Introduction to basic Java syntax: Variables,	- Setting up the Java development environment Install JDK and set up an IDE such as VScode - Write your first java program "Hello, World!"

			data types, operators, control statements	
2	1,2,3	1,2,3	Classes and Objects Understanding classes and objects. Defining classes and creating objects in Java. instance variables, methods, constructors. Understanding <i>this</i> keyword. S - Single Responsibility Principle Definition, intent and benefits; Example of violation Example of correct SRP	- Create a class representing a real-world entity. - Instantiate objects and manipulate object attributes. - Demonstrate constructor usage and method calling. - Create simple Java programs and implement SRP by dividing responsibilities across multiple classes. - Given sample code, check compliance with SRP. - Provided a simple class with a violation of SRP, refactor the code to adhere to SRP.
3	1,2	1,2,3	Encapsulation and Access Modifiers - Introduction to encapsulation. - Access modifiers: public, private, protected, default. - Getter and setter methods. - Why encapsulation is important in OOP.	Implement encapsulation by creating classes with private variables and public getter/setter methods.
4	1,2,3	1,2,3	Inheritance - Understanding inheritance. <i>extends</i> keyword and method overriding. <i>super</i> keyword and constructor chaining. - Types of inheritance OCP - Open/Closed Principle - Definition, intent and benefits; - Example of violation - Example of correct OCP	- Implement inheritance. - Demonstrate method overriding and using the super keyword. - Provided a simple class with a violation of OCP, refactor the code to adhere to OCP using inheritance. - Given sample code, check compliance with OCP.
5	1,2,3	1,2,3	Polymorphism - Understanding polymorphism - Method overloading and method overriding. - Compile-time vs runtime polymorphism. - Interfaces and abstract classes. Reinforce OCP with polymorphism Using inheritance and interfaces to achieve OCP.	- Demonstrate method overloading and overriding with examples. - Implement interfaces and abstract classes - Use polymorphism to call methods on objects of different classes. - Code programs demonstrating OCP with polymorphism.
6	1,2,3	1,2,3	Abstraction - Introduction to abstraction in Java. - Abstract classes vs interfaces. - When to use abstract classes and interfaces. - Real-world examples of abstraction. LSP - Liskov Substitution Principle	- Create an abstract class with abstract methods and a concrete subclass. - Provided a simple class with a violation of LSP, refactor the code to adhere to LSP.

			- Definition, intent and benefits; - Example of violation - Example of correct LSP	
7	2,4	1,2,3	Exception Handling - Introduction to exception handling. - try, catch, finally, throw, and throws keywords. - Types of exceptions: Checked vs unchecked exceptions. - Best practices for handling exceptions.	- Write Java programs to handle exceptions. - Create custom exception classes and use them in a program.
8	2	1,2,3	Collections Framework - I - Introduction to Java Collections Framework. - List, Set, Map, and Queue interfaces.	Create programs to manipulate collections
9	2,3	1,2,3	Collections Framework - II - Working with ArrayList, and LinkedList. - Iterators & enhanced for loop ISP: Interface Segregation principle - Definition, intent and benefits; - Example of violation - Example of correct ISP	- Create programs to manipulate collections - Design an interface that builds different types of toys and check Compliance with ISP.
10	2,4	1,2,3	Multithreading - Introduction to multithreading. - Creating threads using Thread class and Runnable interface. - Thread synchronization and thread lifecycle.	- Create simple multi-threaded programs. - Demonstrate thread synchronization using synchronized keyword.
11	2,4	1,2,3	File Handling and Annotations - Introduction to file handling in Java. - Reading from and writing to files - Serialization & deserialization Java Annotations - What are annotations in Java? - Commonly used annotations (@Override, @Deprecated, @SuppressWarnings) - Custom annotations and their use cases	Write a Java program that - reads a text file and prints its contents to the console - creates a new text file and writes some content to it. - append data to an existing file handle exceptions properly. - Write a Java program that uses annotations to store file paths or configurations.
12	3,5	1,2,3	Database Connectivity in Java - Overview of JDBC (Java Database Connectivity): - What is JDBC and why is it used? - Basic components of JDBC: Driver, Connection, Statement, ResultSet. - Steps for Database Connectivity in Java - JDBC API and Database Operations Dependency Inversion Principle (DIP) - Definition, intent and benefits; - Example of violation	- Performing CRUD Operations using JDBC - Applying DIP to Database Connectivity

			▪ Example of correct DIP	
13	1,2,3,4,5	1,2,3	Mini Project	▪ Create a simple java application such as To Do list

4 CIE Assessment Methodologies

Sl. No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

5. SEE – Practice Assessment Methodologies0

Sl. No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

6. CIE Theory Test model question paper

Program	Information Science and Engineering			Semester – IV	
Course Name	Object Oriented Programming with Java			Marks	50
Course Code	25CS42I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	Questions		Cognitive Level	Course Outcome	Marks
Section - 1					
1	a) Explain the concept of OOP and its key objectives.		L2	1	5
	b) Describe the different control statements used in Java.		L2	1	5
	c) Explain how Single Responsibility Principle (SRP) is violated in the below code and refactor the code to adhere to SRP.		L3	2	15
	<pre>class OrderProcessor { public void processOrder(Order order) { // Process the order } public void generateInvoice(Order order) { // Generate invoice for the order } public void sendEmailConfirmation(Order order) { // Send email confirmation for the order } }</pre>				
2	a) Briefly explain importance of SOLID principles.		L2	1	5
	b) <pre>class Employee { private String name; public int age; protected double salary; String department; }</pre>		L2	1	5
	Explain the use of different access modifiers in the Employee class.				
	c) Explain how Open/Close Principle (SRP) is violated in the below code and refactor the code to adhere to OCP.		L3	2	15
	<pre>class AreaCalculator { double calculateArea(String shapeType, double dimension) { if (shapeType.equals("circle")) { return Math.PI * Math.pow(dimension, 2); } else if (shapeType.equals("square")) { return Math.pow(dimension, 2); } else { throw new IllegalArgumentException("Shape not supported"); } } }</pre>				

Section - 2				
3	a) Explain with an example how encapsulation is implemented in Java using getter and setter methods.	L2	1	5
	b) Write a Java program to demonstrate the use of constructors and the this keyword.	L2	1	5
	c) Refactor the following code to adhere to encapsulation principles:	L3	2	15
	<pre> class Student { public String name; public int age; public void displayDetails() { System.out.println("Name: " + name + ", Age: " + age); } } public class Main { public static void main(String[] args) { Student student = new Student(); student.name = "John"; // Direct access to the fields student.age = 20; // Direct access to the fields student.displayDetails(); } } </pre>			
4	a) Write a Java program to demonstrate the types of inheritance (single and multilevel).	L2	1	10
	c) Refactor the following real-world example of a banking application to demonstrate the principle of abstraction.	L3	2	15
	<pre> class SavingsAccount { public void deposit(double amount) { System.out.println("Depositing \$" + amount + " into Savings Account"); } } class CurrentAccount { public void deposit(double amount) { System.out.println("Depositing \$" + amount + " into Current Account"); } } public class Main { public static void main(String[] args) { SavingsAccount savings = new SavingsAccount(); savings.deposit(500); CurrentAccount current = new CurrentAccount(); current.deposit(1000); } } </pre>			
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Signature of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

7. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	IV
Course Name	Object Oriented Programming with Java			Marks	50
Course Code	25CS42I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
1. Suppose you are tasked with developing a simple Library Management System in Java that demonstrates key Object-Oriented Programming (OOP) principles and SOLID principles. The system should allow the following operations: a) Adding books to the library. b) Borrowing and returning books. c) Displaying available books. d) Handling exceptions such as borrowing unavailable books or returning books not in the library.				1,2,3,4	50
Scheme of assessment(Can be altered as per week wise concepts by the Faculty) 1. Writing the program by identifying and proper use of concepts - 30 a. <i>Defining classes</i> – 5 b. <i>Encapsulation</i> – 5 c. <i>Inheritance and Polymorphism</i>- 10 d. <i>SRP/OCP adherence</i>– 5 e. <i>Exception Handling</i> – 5 2. Final Integration and execution - 20					
TotalMarks					50

Signature of the Course Coordinator

Signature of the HOD

8. SEE–Model Theory Question Paper

Program	Information Science and Engineering	Semester	IV	
Course Name	Object Oriented Programming with Java	Marks	50	
Course Code	25CS42I	Duration	90 Min	
Note: Answer any one full question from each section. Each full question carries equal marks.				
Q No	Questions	Cognitive Levels	Course Outcomes	Marks
Section -1				
1	Illustrate the concept of encapsulation with an example in Java. Explain how access modifiers (private, protected, public) help achieve encapsulation.	L2	1	10
2	Differentiate between instance variables, class variables, and local variables in Java. Write a program to demonstrate	L2	1	10

	their declaration, initialization, and scope.			
Section -2				
3	Create a class to represent a student with attributes for name, roll number, and marks. Write methods to calculate the grade and display the student details.	L3	2	10
4	Implement a Java program to demonstrate polymorphism using method overriding with examples like shape area calculation.	L3	2	10
Section -3				
5	Write a program to read an integer array from a file. Handle exceptions like file not found or invalid data format.	L2	3	10
6	Write a program to handle file reading and parsing errors in a CSV processing system. Include exception handling for scenarios like missing files and invalid formats.	L2	3	10
Section -4				
7	Write a Java program that simulates a parking lot with a limited number of parking spaces using multi-threading. Each car is represented by a thread, and the car tries to park in the lot. If a space is available, it parks. If no space is available, the car waits until a space is free. Requirements: <i>Parking Lot with Limited Spaces:</i> The parking lot has a fixed number of spaces (e.g., 3). <i>Car Class:</i> Each car is a thread and tries to park. <i>Main Class:</i> The main method should create a parking lot with a fixed number of spaces and start multiple car threads (e.g., 5 cars).	L3	4	10
8	Write a Java program to simulate a traffic light system using multi-threading. Let each thread represents a traffic light at an intersection with a random time interval for red, yellow, and green signals.	L3	4	10
Section -5				
9	Write a Java code for a student database using JDBC. Implement only read operation. Use ArrayList to hold the results from a database query, and iterate using enhanced for-loops.	L3	5	10
10	Design a Java program to manage an online store inventory using JDBC. Use collections like HashMap to store product details and demonstrate search operation.	L3	5	10

Signature of the Examiner

2) Signature of the Examiner

9. Rubrics for Assessment of Activity(QualitativeAssessment)

Sl. No.	Dimension	Excellent (10)	Good (8)	Satisfactory (6)	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
TOTAL						40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

10. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	i3/i5	30
02	Internet Connection	100mbps	



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	4
Course Name	Data Analysis and Algorithm Design	Type of Course	Integrated
Course Code	25IS43I	Contact Hours	7 Hours per week
Teaching Scheme	3:0:4	Credits	5
CIE Marks	50	SEE Marks	50 (Practice)

1. Rationale:

This course aims to equip learners with the essential tools and skills necessary to handle data efficiently, apply advanced algorithm design strategies, and evaluate their performance in terms of time and space complexity. By integrating NumPy, Pandas, Matplotlib, and algorithmic concepts, the course prepares students to solve complex real-world problems and perform data-driven tasks efficiently.

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

CO-01	Utilize appropriate Python modules and functions to perform efficient numerical computations and solve data-driven problems.
CO-02	Analyse given data and visualise insights using right python modules
CO-03	Select and apply appropriate algorithm design strategies, to solve computational problems.
CO-04	Analyse space and time complexities of the algorithm to evaluate their efficiency.
CO- 05	Identify and resolve issues in the implementation of algorithmic solutions to ensure correctness and efficiency

3. Course Content

Wee k	CO	P O	Theory	Practice
1	1		NumPy Arrays: Overview, Array Creation Methods: array(), arange(), linspace(), zeros(), ones(), empty() Random Number Generation: Creating random arrays with: np.random.rand(), np.random.randn(), np.random.randint() Array Indexing and Slicing Setting the random seed with np.random.seed() Aggregation functions: np.sum(), np.mean(),	Implement a python code to • Create and manipulate 1D and 2D arrays. • Write code to display array attributes such as shape, size, and dimension. • Create arrays of evenly spaced values using np.arange() and np.linspace(). • Generate random arrays of integers and floats. • Retrieve specific rows, columns, or subsets of a 2D array. • Apply Boolean indexing to filter elements based on conditions.

			np.median(), np.std(), np.var())	● Modify a given array using slicing and indexing. ● Compute statistical measures (mean, median, standard deviation) for a dataset
2	1		NumPy Arrays contd. Array Reshaping: Changing the shape of arrays: np.reshape(), np.resize() Flattening arrays: np.flatten(), np.ravel() Transposing arrays: np.transpose(), np.swapaxes() Stacking arrays: np.hstack(), np.vstack() Splitting arrays: np.split(), np.hsplit(), np.vsplit() Broadcasting in NumPy Element-wise Operations Using ufuncs: np.add(), np.subtract(), np.multiply(), np.divide()	Implement a python code to ● Reshape a 1D array into 2D and vice versa. ● Flatten a 2D array and explore the difference between np.flatten() and np.ravel(). ● Combine arrays horizontally and vertically, and split them into smaller arrays. ● Perform element-wise operations on two arrays ● Perform arithmetic operations between arrays of different shapes using broadcasting.
3	2		Getting Started with pandas ● Differences between pandas and Excel/numpy. Series and DataFrame Basics ● Creating a Series and exploring its attributes (e.g., .values, .index). ● Creating DataFrames (from dictionaries, lists, numpy arrays). ● DataFrame attributes (.shape, .columns, .dtypes, .head(), .tail()). Data Selection and Manipulation Indexing and Selection ● Selecting rows and columns using .loc[] and .iloc[]. ● Boolean indexing to filter rows based on conditions. Modifying Data ● Adding new columns, Updating column values. ● Dropping columns and rows (.drop()). Handling Missing Data ● Detecting missing values (.isnull(), .notnull()). ● Filling missing values (.fillna()). ● Dropping missing data (.dropna()).	Implement a python code to ● Create and explore a Series for a small dataset (e.g., sales or student scores). ● Create DataFrames from dictionaries and NumPy arrays. ● Use .head() and .tail() to inspect the structure of the DataFrame. ● Retrieve specific rows and columns from a DataFrame using .loc[] and .iloc[]. ● Apply filters to a dataset using Boolean indexing (e.g., selecting rows based on conditions). ● Modify the index of a DataFrame. ● Add and delete columns in a DataFrame. ● Handle missing values in a dataset by filling or dropping them. ● Sort a DataFrame based on specific columns or index.
4	2		Pandas contd. Data Analysis and Grouping: Sorting and Ranking ● Sorting values by rows or columns (.sort_values(), .sort_index()). ● Ranking data using .rank().	Implement a python code to ● Compute basic statistics (sum, mean, etc.) for a dataset. ● Group a dataset by a specific column and compute aggregated

			Aggregation and Descriptive Statistics - Summary statistics (.mean(), .median(), .std(), .describe()). - Aggregation methods (.sum(), .count(), .min(), .max()). Grouping Data - Grouping data with .groupby(). - Aggregating grouped data (.size(), .agg(), .apply()).	- statistics for each group. - Use multiple aggregation functions on grouped data.
5	1 2		Functional Programming Built-in higher-order functions in Python: map(), filter(), reduce(). Lambda functions: - Syntax and use cases for anonymous functions. - Difference between lambda functions and regular functions.	- Given a list of numbers [1,2,3,4,5], use a lambda function with the map() function to square each element of the list. Return and display the new list of squared values. - Given a list of integers [12,15,8,24,30], write a lambda function with filter() to select only the numbers that are divisible by 5. Display the filtered list. - Using the list [1,2,3,4,5], write a lambda function that uses reduce() to compute the product of all elements in the list. Display the final result.
6	2		Data visualization with python - Importance of Data Visualization - Types of visualizations: Line plots, bar charts, histograms, scatter plots, etc. - Overview of Python visualization libraries: Matplotlib, Seaborn, and Plotly. Matplotlib: Structure of Matplotlib-Figure, Axes, and Plotting. Plot customization: - Titles, labels, and legends. - Adjusting axis limits and scales (logarithmic vs linear). - Styling plots (colors, markers, line styles).	Install and set up Matplotlib and Seaborn. 1.Create basic plots using Matplotlib: - Line plot - Bar chart - Scatter plot 2.Create a customized line plot, bar chart, and scatter plot. - Modify titles, labels, and legends. - Change plot colors and styles.
7	2		Seaborn: Introduction to Seaborn and its advantages over Matplotlib. Seaborn's default aesthetic style. Types of plots in Seaborn: Boxplot, violin plot, pairplot, heatmap, etc. Customizing Seaborn plots: - Changing plot colors, palettes, and styles. - Adjusting plot sizes and axis labels. - Plotting categorical data (e.g., barplot, countplot).	1.Create basic plots using Seaborn: - Boxplot - Histogram with KDE - Pairplot Work with Seaborn's built-in datasets (e.g., tips, iris) 2.Create and customize Seaborn plots: - Modify plot colors using Seaborn color palettes. - Customize axis labels and titles. - Create bar plots and count plots

				for categorical data.
8	3 4 5		Introduction to algorithm design paradigms. Importance of algorithm design and analysis. Algorithm efficiency: Time complexity and space complexity. Asymptotic Analysis • Types of Asymptotic Notation • Steps to Perform Asymptotic Analysis • Best, worst, and average-case time complexities. • Common Time Complexities in Asymptotic Analysis Problem-solving strategies: Brute force, Decrease and conquer, Divide and conquer, greedy, Dynamic programming.	1. Write a program to calculate the sum of the first n numbers using: • A loop ($O(n)$) • A formula ($O(1)$) Asymptotically analyse the space complexity and time complexity of both approaches by varying n value and plot the graph using asymptotic notations. 2. Write a program to calculate the Fibonacci sequence • Recursively and • Iteratively Asymptotically analyse the space complexity and time complexity of both approach by varying n value and plot the graph using asymptotic notations
9	3 4 5		Brute force approach • Bubble sort, • Selection Sort, • Linear Search.	1. Implement Linear Search compute space and time complexities. Plot the graph using asymptotic notations. 2. Implement Bubble, Selection sorting algorithms compute space and time complexities. Plot the graph using asymptotic notations
10	3 4 5		Decrease and conquer: Insertion Sort. Greedy – Concepts only. Dynamic programming: Fibonacci sequence.	1. Implement insertion sort algorithms compute space and time complexities. Plot graph using asymptotic notations. 2. Implement Fibonacci sequence with dynamic programming. Plot graph using asymptotic notations
11	3 4 5		Divide and conquer - Merge Sort, Quick Sort, Binary search. Backtracking – Concepts only	1. Implement Merge and quick sorting algorithms. compute space and time complexities. Plot graph using asymptotic notations and compare both solutions. 2. Implement Binary Search using recursion. Compute space and time complexities. Plot graph using asymptotic notations.
12	3 4 5		Introduction to Graphs Graph definitions, terminology. Types of graphs. Represent graphs in different ways: (adjacency matrix, adjacency list, and edge list).	Implementing BFS and DFS BFS Implementation: ○ Use a queue to traverse the graph level by level.

			Graph Traversal Algorithms Breadth-First Search (BFS) Depth-First Search (DFS)	DFS Implementation: ○ Implement both recursive and iterative approaches for DFS.
13	3 4 5		Introduction to Hashing. Hashing - Perfect hashing functions. Hash table Hash Functions, Operations, Hash collision, Application.	Implement Password Verification System Using Hashing(use Python's hashlib module, which provides hashing functions like SHA-256 to demonstrate secure password handling.)

4. Learning Resources:

Sl No	References
1	"Data Analysis with Python" by David Taieb
2	"Python for Data Analysis" by Wes McKinney
3	"Algorithms: Design and Analysis" by Tim Roughgarden
4	https://docs.python.org/3/tutorial/datastructures.html
5	https://www.geeksforgeeks.org
6	https://www.w3schools.com
7	https://www.tutorialspoint.com
8	https://www.khanacademy.org
9	https://www.datacamp.com
10	https://www.programiz.com

5. CIE Assessment Methodologies

Sl	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1 TheoryTest	4	90	50	Average of all CIE=50 Marks
2.	CIE-2 Practice Test	7	180	50	
3	CIE-3 TheoryTest	10	90	50	
4.	CIE-4 Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 Marks

6. SEE – Practice Assessment Methodologies

Sl	SEE – Practice Assessment	Duration (minutes)	Max Marks	Min marks to pass
1.	Semester End Examination- Practice	180	50	20

7. CIE Theory Test model question paper

Program	Information Science and Engineering			Semester 4	
Course Name	Data Analysis and Algorithm Design			Test	
Course Code	25IS43I	Duration	90 min	Marks	50
Name of the Course Coordinator:					

Note: Answer any one full question from each section. Each full question carries equal marks.				
Q.No	Questions	CL	CO	Marks
Section – 1				
1	Problem Statement: Weather Data Analysis A meteorological department has collected temperature data for a city over a year. The data is stored as a 2D NumPy array, where each row represents a month's data, and each column represents the daily temperatures for that month. Your task is to analyze the temperature data using NumPy to answer the following questions: ○ Load the temperature data (a 12 x 31 NumPy array where rows represent months and columns represent days). ○ Verify the shape of the array to ensure it matches the expected dimensions. ○ Calculate the average temperature for each month. ○ Identify the month with the highest and lowest average temperatures. ○ Determine the day of the year with the highest temperature. ○ Find the day of the year with the lowest temperature.			25
2	Problem Statement: Sales Data Analysis A retail store collects daily sales data for a week from three branches. The data is stored in a 2D NumPy array, where each row represents a day of the week, and each column represents the sales figures for a specific branch. Your task is to analyze the sales data using NumPy to answer the following questions: ○ Load the sales data into a 2D NumPy array. ○ Verify the shape of the array (it should be 7 x 3). ○ Calculate the total sales for each branch over the week. ○ Find the branch with the highest total sales. ○ Calculate the total sales for each day across all branches. ○ Identify the day with the highest total sales. ○ Compute the average daily sales for the entire store. ○ Identify the day with sales closest to the weekly average. ○ Sort the sales data for each branch in ascending order (row-wise). ○ Flatten the array to analyze all sales figures as a single list.			
Section – 2				
3	Problem Statement: Student Grades Analysis A school maintains a record of students' grades for three subjects (Math, Science, and English) in a CSV file. Each row represents a student, and the columns represent the student's name, roll number, and their grades for the three subjects. ○ Load the student grades data into a Pandas DataFrame. ○ Display the first 5 and last 5 rows of the dataset. ○ Print the column names and the number of students (rows). ○ Calculate the average grade for each subject. ○ Find the subject in which students scored the highest average grade.			25

	○ Find students who scored more than 90 in Math. ○ Sort the students by their total marks in descending order. ○ Assign a rank to each student based on their total marks.			
4	Problem Statement: Employee Salary Data Analysis A company maintains an employee salary record in a CSV file. The dataset contains the following columns: Employee ID, Name, Department, Salary, and Joining Date. Each row represents an employee. ○ Load the employee salary data into a Pandas DataFrame. ○ Display the first 5 rows of the dataset. ○ Show the total number of employees and the list of unique departments. ○ Calculate the average salary for the entire company. ○ Find the department with the highest average salary. ○ Identify the employee with the highest salary in the company. ○ Find employees whose salaries are greater than 80,000. ○ List all employees who joined the company before 2015. ○ Calculate the total salary expenditure for each department. ○ Count the number of employees in each department.			
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

8. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	4
Course Name	Data Analysis and Algorithm Design			Test	
Course Code	25IS43I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
Problem Statement: Sales Data Analysis and Visualization Problem Statement: A retail company wants to analyze its sales data for the past year, which is stored in a CSV file. The dataset includes daily sales figures for various product categories (e.g., Electronics, Clothing, and Groceries). The company wants to perform the following tasks to understand its sales performance and visualize key metrics: 1. Load the sales data from a CSV file into a Pandas DataFrame. 2. Clean the data by handling any missing values or outliers. 3. Use NumPy to perform calculations such as calculating the daily sales growth and determining the total sales for each month. 4. Group the data by product category and calculate the average sales per category, as well as the total sales for each product. 5. Plot key visualizations using Matplotlib, such as: ○ A line plot to show the sales trend over time. ○ A bar chart to compare the total sales per product category.					50

o A pie chart showing the market share of each product category		
Scheme of assessment (customize scheme as per requirement of your problem statement)		
a) Program Design and Conceptual Clarity (Clear identification of the key concepts, Explanation of the logic and methodology used and organization of the program.) - 10		
b) Implementation and Execution - 30		
c) Best Practices (Code Readability and Error Handling)- 10		
TotalMarks		50

Signature of the Course Coordinator

Signature of the HOD

9. SEE- Model Practice Question Paper

Program	Information Science and Engineering		Semester	4
Course Name	Data Analysis and Algorithm Design	Course Code : 25IS43I	Duration	180 min
Questions			CO	Marks
Social Media Engagement Analysis Problem: You have data from a social media platform showing User_ID, Post_ID, Likes, Comments, Shares, and Date. • Import the data into a Pandas DataFrame. • Calculate the total number of interactions (likes, comments, shares) for each post. • Group the posts by date and calculate the average engagement (total interactions per post). • Implement a Divide and Conquer algorithm to segment users into different engagement levels based on their total interactions. • Create a bar chart to visualize total interactions by post. • Plot a time series graph of user engagement over time.				50
Scheme of assessment: a)Load and analyse data (10 Marks) b)Algorithm selection and design((10 marks) c)Implementation(20 marks) c)Visualization(5 marks) d)Presentation(5 marks)				
Total Marks				50

Signature of the Examiner

2) Signature of the Examiner

10. Rubrics for Assessment of Activity(Qualitative Assessment)

Sl. No.	Dimension	Excellent (10)	Good (8)	Satisfactory (6)	Needs Improvement	Score
---------	-----------	----------------	----------	------------------	-------------------	-------

					(4 or below)	
		9-10	7-8	5-6	<=4	
1	Understanding of Concepts	Demonstrates thorough understanding of underlying concepts and explains them confidently.	Shows good understanding of concepts with minor gaps.	Demonstrates basic understanding but struggles with deeper applications.	Lacks understanding of concepts or is unable to explain them clearly.	8
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
					TOTAL	40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

11. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	i3 or i5	30
02	Internet Connection	100 Mbps	



Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	Information Science and Engineering	Semester	4
Course Name	Web Development	Type of Course	Integrated
Course Code	25IS44I	Contact Hours	7 per week
Teaching Scheme	3: 0:4	Credits	5
CIE Marks	50	SEE Marks	50 (Practice)

1. Rationale: The course aims to equip students with a comprehensive understanding of user interface (UI) and user experience (UX) design principles, alongside practical front-end development skills. By incorporating a design thinking approach, the course empowers students to create functional, visually appealing, and user-friendly web applications.

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

CO1	Create visually appealing user interfaces using design tools.
CO2	Transform UI designs into functional web interfaces using HTML and CSS.
CO3	Develop a responsive webpage with interactive functionality using JavaScript.
CO4	Develop reusable React components and implement component interaction
CO-5	Develop a fully functional Single Page Application (SPA) by integrating state management, routing, and UI components

3. Course Content

Week	CO	PO	Theory	Practice
1	1		Introduction to UI/UX Design - Overview of UI/UX; Differences and their roles in web development; - Principles of UI/UX Design; UI elements (Buttons, Text Fields, Dropdowns, Checkboxes and Radio Buttons, Sliders, Date Pickers, Cards, Navigation Bars, Progress Bars, Notifications/Alerts, etc) - Examples of good and bad UI/UX. - Color theory & Typography	Learn wire framing and prototyping using Figma or similar design tools. Gain hands-on experience with the tool's interface, including its key features and functionalities, to create effective and user-friendly design mockups
2	1		Design thinking - 5 stages Integrating Design Thinking with UI/UX design	- Create a simple wireframe for a login or e-commerce page. - Create wireframes to visualize layouts and structures. Case Study - Problem: Users find it challenging to navigate a banking app to transfer money.

				Problem: Users frequently abandon their orders during the checkout process in a food delivery app. Use Design Thinking to redesign UI/UX to solve above problems
3.	2		HTML Basics 1. Structure of an HTML document. Basic Tags, Elements, Attributes and comments Text formatting tags : <p>; <h1> -<h6>; , <i>, <u>, , ,,<hr> - Lists: Ordered and Unordered ; - Links: <a>; - Images: ; - block & inline elements: <div> and 	Prepare and get familiarize with Your Development Environment like VScode or similar IDE Practice using HTML tags to structure and design user interfaces, focusing on creating layouts, Refer table for suggestive exercises
4	2		1. Tables - <table>, <tr>, <td>, <th> 2. Forms and Input Elements Form structure: <form> Input fields: <input>, <textarea>, <button> Dropdowns and checkboxes: <select>, <option>, <input type="checkbox"> Radio buttons: <input type="radio"> Media elements.	Practice using HTML tags to structure and design user interfaces, focusing on creating layouts, Refer table for suggestive exercises
5	2		1. What is CSS? Why is it important? Different ways to apply CSS: Inline, Internal, External. 2. Understanding the CSS syntax: Selectors, properties, and values. 3. Color; Background; Fonts and Text Styling	Create a simple HTML file and apply inline, internal, and external CSS. Refer table for suggestive exercises
6	3		What is JavaScript? Role of JavaScript in web development. ES6 features; Embedding JavaScript in HTML: <script> tag. - Variables (var, let, const) and data types. - Operators and expressions. Conditional statements (if-else, switch). - Loops (for, while, do-while). Loop control statements: break and continue.	Refer table for suggestive exercises
7	3		- Functions and scope - DOM Manipulation: What is DOM? Accessing elements, modifying elements, creating and adding elements, removing elements - Event handling : event and types [Mouse events, keyboard event, form events, windows events] Event Listeners Event propagation and delegation	Refer table for suggestive exercises Selecting and modifying HTML elements. Event handling (e.g., click, mouseover). Basic form validation

8	3	Arrays and Objects: Arrays: Create, Indexing, array methods, ES6 enhancements for arrays Objects: create, accessing properties, object methods ES6 Enhancements for Objects	Refer table for suggestive exercises
9	3	Introduction to JSON What is JSON? Structure; Syntax Rules; Common Use Cases Working with JSON in JavaScript Converting JavaScript Objects to JSON Converting JSON to JavaScript Objects Handling JSON in APIs Working with JSON in Web Applications	Create and Parse JSON
10	4	Introduction to React: How traditional websites worked Evolution of websites What is React? How react app works? Basic Syntax of React .js React components and props	Setting up the environment- install required tools – vscode node.js Create your first React app using vite – create – run - stop Create your first component
11	4	React Component Interaction State and lifecycle methods. React Hooks: What are React Hooks? Hook – purpose, syntax and basic usage [useState, useEffect, useContext, useReducer] React Forms and React Styles	Learn state management with React hooks Create and interact with react forms
12	5	Introduction to React Router Basic Routing Dynamic Routing Nested Routing Redirects and Programmatic Navigation Protected Routes	Install React Router and Setup Dynamic Routing with Parameters Learn how to redirect users based on certain conditions using the <Redirect /> component. Learn how to create protected routes that only allow authenticated users to access certain pages. Create a login system with protected routes
13	5	Introduction to SPA: What is a Single Page Application; Characteristics; Advantages; Basic Structure of a React SPA; Building a Basic SPA with React.	Build simple REACT SPA.

4 .Learning Resources:

1. <https://www.geeksforgeeks.org/principles-of-ui-ux-design/>
2. <https://www.elinext.com/services/ui-ux-design/trends/typography-color-principles-in-ui-ux-design/>
3. <https://medium.com/design-bootcamp/using-colors-and-typography-wisely-ui-ux-design-strategies-to-capture-user-attention-52b0eb5d35f3>
4. <https://chatgpt.com/>
5. https://www.youtube.com/watch?v=c9Wg6Cb_YIU
6. <https://youtu.be/dXQ7IHkTiMM?si=BTzfn7b-DCHgb2he>

Books

1. "Don't Make Me Think" by Steve Krug
2. "The Design of Everyday Things" by Don Norman

Online Courses

Coursera

1. "UI/UX Design Specialization" by California Institute of the Arts: Comprehensive introduction to UI/UX design.
2. "Introduction to User Experience Design" by Georgia Tech: A beginner-friendly course on UX fundamentals.

Udemy

1. "User Experience Design Fundamentals" by Joe Natoli: Beginner-friendly with practical examples.
2. "Learn Figma – UI/UX Design Essential Training": Focused on wireframing and prototyping using Figma.

LinkedIn Learning

1. "UX Design: From Concept to Prototype": Covers wireframing and prototyping tools like Sketch and Adobe XD.

<https://www.springboard.com/blog/design/design-thinking-process/>

<https://www.geeksforgeeks.org/five-phase-model-for-design/>

<https://www.youtube.com/watch?v=4nTh3AP6knM>

https://www.youtube.com/watch?v=r0VX-aU_T8

"The Design Thinking Playbook" by Michael Lewrick, Patrick Link, and Larry Leifer

"Design Thinking: Understanding How Designers Think and Work" by Nigel Cross

"*Learning Design Thinking*": Focuses on applying design thinking in real-world scenarios.

"*Design Thinking Guide for Successful Professionals*": Practical and industry-oriented.

Stanford d.school Resources

HBR Design Thinking

"Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins

"HTML5: The Missing Manual" by Matthew MacDonald

freeCodeCamp HTML : "*HTML and HTML5*":

Codecademy HTML : "*Learn HTML*":

Coursera HTML : "HTML, CSS, and JavaScript for Web Developers" by Johns Hopkins University

Udemy HTML : "Build Responsive Real World Websites with HTML5 and CSS3"

<https://www.youtube.com/watch?v=IWcUJLUAO2A>

https://www.youtube.com/watch?v=6OhMbf2v_jl

<https://react.dev/learn/describing-the-ui>

5. Suggested Programs

week	suggestive program/activity
1	Wireframe Login page participation registration form for an event, etc
2	Conduct warmup activities to ignite Design Thinking ▪ Empathize: Understanding user behaviors, needs, and pain points. ▪ Define: Clearly articulate the user problem ▪ Ideate: Brainstorm creative solutions to the problem ▪ Prototype: Create Visual Representations - Create wireframes to visualize layouts and structures. ▪ Test: Validate the prototype by gathering user feedback.

3	"Create Your First HTML Document" create a basic HTML document to demonstrate your understanding of fundamental HTML structure and tags. The document should meet the following requirements: - Include a <head> section with a title for the webpage. - Add a main heading (<h1>) to introduce the page with a welcoming message. - Include a paragraph (<p>) describing the purpose of the page. - Create an unordered list () with at least three list items () explaining basic facts about HTML. - Add a hyperlink (<a>) to an external website, ensuring it opens in a new tab. Format the text to enhance the visual appeal and readability
4	Design a product detail page for an e-commerce website. The page should display the product name, image, description, price, and availability. There should also be options to select the quantity and add the product to the shopping cart. Design a login and registration page for a website. The login form should include fields for email and password, while the registration form should include additional fields such as username, email, password, and confirm password. Design a user profile page where users can view and edit their personal details such as name, email, phone number, profile picture, and address. The page should have a navigation bar at the top and a sidebar with links to other user settings (e.g., Account Settings, Privacy Settings). Design a product listing page for an e-commerce website. The page should display a list of products with the following details: product name, price, product image, description, and a "Add to Cart" button. The page should be responsive and adapt to different screen sizes (desktop, tablet, and mobile). Project - Create a basic webpage that includes: - A title and headings. - Paragraphs and images. - A table and a form. - Links to navigate between sections or pages. Adding a video and audio file to your webpage.
5	Style basic elements like <p>, <h1>, and <div> Create an HTML page and apply different selectors to style elements. Experiment with background colors and text colors using various color formats. Style paragraphs and headings using different fonts and text properties. Enhance the visual appeal of the UI design in the above section by applying CSS
6 & 7	Write and execute your first JavaScript program. Practice using console.log() for debugging. Write code to compare two numbers and display the larger one Calculate the Total Bill (Shopping Cart) Create a program that calculates the total price of items in a shopping cart. The cart is represented as an array of prices. Use a loop to iterate through the array and calculate the total. Create a Personal Expenses Tracker Design and build a simple web-based personal expenses tracker that allows users to:

Add an expenses (description, amount, and date).

View the list of expenses.

Calculate and display the total amount spent.

HTML

- a. Create a form with the following fields:
 - Description (text input)
 - Amount (number input)
 - Date (date input)
- b. A button to add the expense to the list.
- c. Add a table or list to display the expenses.

CSS

- a. Style the form to make it visually appealing.
- b. Add hover effects on the "Add Expense" button.
- c. Use a color scheme to distinguish the form, expense list, and total amount display.

JavaScript

- a. Validate the form fields (e.g., all fields must be filled).
- b. Add functionality to:
- c. Add the entered expense to the list when the button is clicked.
- d. Calculate and update the total amount after each addition.

Create a To-Do List Application

Design and build a simple web-based to-do list application that allows users to:

- Add tasks to a list.
- Mark tasks as completed.
- Delete tasks from the list.

HTML

- a. Create a form with a single input field to enter a task description.
- b. Add a button to add the task to the list.
- c. Create a section to display the list of tasks, with each task having:
 - A checkbox to mark it as completed.
 - A delete button to remove it from the list.

CSS

- a. Style the form and task list to make the application visually appealing.
- b. Use different styles for completed tasks (e.g., strikethrough text or a light background color).
- c. Add hover effects on buttons and smooth transitions for task updates.

JavaScript

Add functionality to:

- a. Validate that the input field is not empty before adding a task.
- b. Add the task to the list dynamically.
- c. Mark tasks as completed when the checkbox is clicked.
- d. Remove tasks from the list when the delete button is clicked.

Create a Temperature Converter

Design and build a simple web-based temperature converter that allows users to:

- Input a temperature in Celsius or Fahrenheit.
- Convert the temperature to the other unit.
- Display the result dynamically on the page.

HTML

Create a form with:

- An input field for the temperature value.
- A dropdown menu to select the unit (Celsius or Fahrenheit).
- A button to perform the conversion.
- Add a section to display the converted temperature.

	CSS - Style the form and result section to make the application visually appealing. - Add a consistent color scheme and spacing for a clean layout. - Use hover effects for the button JavaScript Add functionality to: - Validate that the input field contains a valid number. - Perform the conversion based on the selected unit: - Celsius to Fahrenheit: $(C \times 9/5) + 32$ - Fahrenheit to Celsius: $(F - 32) \times 5/9$ - Display the result dynamically on the page. Create a Currency Converter Design and build a simple web-based currency converter that allows users to: - Input an amount in one currency. - Select the currencies for conversion (e.g., USD to EUR). - Display the converted amount dynamically based on a fixed exchange rate. HTML Create a form with: - An input field for the amount. - Two dropdown menus to select the source currency and target currency. - A button to perform the conversion. - Add a section to display the converted amount. CSS - Style the form and result section for a professional look. - Add a consistent color scheme and spacing. - Use hover effects for the button and smooth transitions for dynamic updates. JavaScript Add functionality to: - Validate the input to ensure it is a positive number. - Use predefined exchange rates to calculate the conversion. For example: - USD to EUR: 1 USD = 0.85 EUR - EUR to USD: 1 EUR = 1.18 USD - USD to INR: 1 USD = 74.50 INR - Add more currencies as needed. - dynamically display the converted amount.
8	Create a simple contact list to store contact information and perform operations like searching and adding new contacts. Design a simple system to manage the items in the shopping cart of an online store using arrays and objects. The system should allow users to add, remove, and update items in the cart, calculate the total cost, and display the cart's contents. Create a system to manage movie rentals. Each movie has a title, genre, rental price, and availability status. The system should allow users to add and remove a movie, update movie info, display all movies and rent a movie. Develop a system to manage tasks for a project. Each task has a title, description, priority, and completion status. The system should allow following operations: - Add Task: Add a new task to the system. - Remove Task: Remove a task by taskID. - Update Task Info: Update the title, description, priority, or status of a task. - Mark Task as Completed: Mark a task as completed. - Display All Tasks: Display the list of all tasks with their titles, priorities, and statuses. - Filter Tasks by Priority: Display tasks with a specific priority level.

9	Create and Parse JSON - Write a JSON string for a "Book" with attributes like title, author, price, and genres. Parse it into an object and access its properties. - Create a JSON object for a "Student" with nested "Subjects" and "Scores". Parse it and calculate the average score. - Convert JSON data (string) into a JavaScript object using <code>JSON.parse()</code>. - Convert a JavaScript object into a JSON string using <code>JSON.stringify()</code>.
10	- Create a Simple Greeting Component - Create a React component called Greeting that accepts a name as a prop and displays a message like "Hello, [name]!". - Build a ColorBox component that displays a colored box. It should take a color prop (e.g., red, blue, green) and render the box in that color. - Create a ProductCard component that displays product information like name, price, description, and an "Add to Cart" button. On clicking the button, log a message "Product added to cart".
11	Counter App: - Create a simple counter app with <code>useState</code>. - Add buttons to increment and decrement the counter. Effect on Mount: - Use <code>useEffect</code> to log a message to the console when the component is mounted. - Add a cleanup function in <code>useEffect</code> to demonstrate component unmounting. Form Validation App: - Create a form with multiple fields (e.g., name, email, password). - Use <code>useState</code> to manage the form data and <code>useEffect</code> to validate the fields. - Show validation messages when the fields are invalid. - Create a LoginForm component with two input fields (username and password) and a submit button. Display a message saying "Welcome, [username]" when the user submits the form. - Create a LoginForm component with two input fields (username and password) and a submit button. Display a message saying "Welcome, [username]" when the user submits the form. - Create a TodoList component where users can add new tasks to a list and remove tasks by clicking a delete button next to each task. - Create a Rating component that displays 5 stars. Clicking on a star sets the rating, and all stars up to that point should be highlighted. - Build a SearchFilter component that allows users to search through a list of items. Display only the items that match the search query. - Create a ShoppingList component that renders an array of items (e.g., "Apples", "Bananas", "Oranges"). Display the items in a <code></code> list. - Create a ToggleButton component with a button that toggles between "ON" and "OFF" states when clicked. - Build a Counter component with a button that increments the count each time it's clicked. - Create a Stopwatch component with start, stop, and reset buttons. Display the time elapsed since the stopwatch was started. - Build a ThemeSwitcher component with a button that toggles between light and dark themes. The background color and text color should change accordingly. - Build a Slider component that allows users to select a value between a defined range (e.g., 0-100) using an input slider. Display the selected value.
12	Basic Routing - Create a simple React app with three pages: Home, About, and Contact. - Use <code>react-router-dom</code> to create routes for each page. - Display a navigation menu with links to each page.

	Output: When you click on the links, the corresponding pages should render without reloading the entire page. Nested Routes ▪ Create a "Dashboard" page that contains two sub-pages: "Profile" and "Settings". ▪ Use nested routes to render these sub-pages inside the Dashboard component. ▪ Display links for "Profile" and "Settings" inside the Dashboard page. Output: Clicking on "Profile" or "Settings" should render the appropriate content inside the Dashboard page. Dynamic Routing ▪ Create a route to display a user profile page. ▪ The URL should contain a user ID (e.g., /profile/:id). ▪ Use useParams to extract the id from the URL and display it on the page. Output: Visiting /profile/1 should display a profile for user 1, and /profile/2 should display a profile for user 2. Redirecting ▪ Create a login page where a user can submit a form. ▪ After a successful login, redirect the user to the "Home" page using the Redirect component. Output: After the login form is submitted, the user should be redirected to the Home page. Programmatic Navigation ▪ Create a button that, when clicked, programmatically navigates to a new route (e.g., from "/home" to "/about"). Output: Clicking the button should redirect the user to the new route. Navigation Bar ▪ Create a navigation bar with links to different pages (Home, About, Contact). Output: The active link should be highlighted based on the current route.
13	E-Commerce Product Catalog Features: Display a list of products with filters (e.g., price, category). Add items to a shopping cart. Update the cart dynamically without refreshing the page. Key Concepts: Dynamic rendering of products. State management for the shopping cart. Online Quiz Application Features: Display multiple-choice questions one at a time. Show results and scores after completion. Include a timer for each question. Key Concepts: Dynamic rendering of questions. State management for answers and scores. Conditional rendering for results and feedback. To-Do List Application Features: Add, edit, and delete tasks.

	Mark tasks as completed. Filter tasks by status (e.g., All, Active, Completed). Key Concepts: Component-based design (e.g., using React or Vue). State management for tasks. Routing (e.g., different views for "All" and "Completed" tasks).
--	---

6. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	Average of all CIE=50 Marks
1.	CIE-1TheoryTest	4	90	50	
2.	CIE-2Practice Test	7	180	50	
3	CIE-3TheoryTest	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 Portfolio evaluation of all the activities through Rubrics	1-13		50	
Total					50 arks

7 SEE – Practice Assessment Methodologies

Sl.No	SEE – Practice Assessment	Duration (minutes)	Max marks	Min marks to pass
1.	Semester End Examination-Practice	180	50	20

8. CIE Theory Test model question paper

Program	Web Development			Semester - 4	
Course Name	Information Science and Engineering			Test	
Course Code	25IS44I	Duration	90 min	Marks	50
Name of the Course Coordinator:					
Note: Answer any one full question from each section. Each full question carries equal marks.					
Q.No	Questions		Cognitive Level	Course Outcome	Marks
Section - 1					
1	a. What are the five stages of design thinking? Briefly describe each stage with examples. - 10 b. Create a simple wireframe for a participation registration form for a seminar. -10		2,3	1	25

	c. Describe core UI/UX design principles. - 5			
2	a. How does empathy play a critical role in UI/UX design? Provide an example of how empathy can influence design decisions. - 5 b. Explain the difference between UI (User Interface) and UX (User Experience) with examples. Design a wireframe for a login page, including features like “Remember Me” and “Forgot Password” links. - 10 c. Explain the concept of progressive disclosure in UI/UX design. How can it be represented in a wireframe? - 10	2,3	1	

Section – 2

3	a. Explain the difference between block-level and inline elements with examples. b. How does the alt attribute in the tag enhance accessibility? c. How would you create a form with the following fields: Name, Email, Password, Gender, and Submit? Write the HTML code. – 10 d. Write HTML code to create following table - 10 <table><thead><tr><th colspan="4">Two - Wheeler</th></tr><tr><th>OEM</th><th>May-23</th><th>May-22</th><th>YOY</th></tr></thead><tbody><tr><td>Hero MotoCorp</td><td>5,08,309</td><td>4,66,466</td><td>9%</td></tr><tr><td>HMSI</td><td>3,11,144</td><td>3,20,844</td><td>-3%</td></tr><tr><td>TVS Motor</td><td>2,52,690</td><td>1,91,482</td><td>32%</td></tr><tr><td>Bajaj Auto</td><td>1,94,811</td><td>96,102</td><td>103%</td></tr><tr><td>Royal Enfield</td><td>70,795</td><td>53,525</td><td>32%</td></tr><tr><td>Suzuki Motorcycle</td><td>64,000</td><td>60,518</td><td>6%</td></tr><tr><td>Total</td><td>14,01,749</td><td>11,88,937</td><td>18%</td></tr></tbody></table>	Two - Wheeler				OEM	May-23	May-22	YOY	Hero MotoCorp	5,08,309	4,66,466	9%	HMSI	3,11,144	3,20,844	-3%	TVS Motor	2,52,690	1,91,482	32%	Bajaj Auto	1,94,811	96,102	103%	Royal Enfield	70,795	53,525	32%	Suzuki Motorcycle	64,000	60,518	6%	Total	14,01,749	11,88,937	18%	2	2	
Two - Wheeler																																								
OEM	May-23	May-22	YOY																																					
Hero MotoCorp	5,08,309	4,66,466	9%																																					
HMSI	3,11,144	3,20,844	-3%																																					
TVS Motor	2,52,690	1,91,482	32%																																					
Bajaj Auto	1,94,811	96,102	103%																																					
Royal Enfield	70,795	53,525	32%																																					
Suzuki Motorcycle	64,000	60,518	6%																																					
Total	14,01,749	11,88,937	18%																																					
4	a. How can you set a default value for a text input field in HTML? b. Create a form to capture patient details during registration. - 10 c. What is the difference between checkboxes and radio buttons in forms? Provide examples. d. Write HTML code to create following form – 10 Please Login To Continue Sign In Sign Up Username or email Password Remember me Forgot Password Sign In or Google Facebook LinkedIn GitHub Why Create an Account? By creating this account, you agree to our Privacy Policy & Cookie Policy.	2	2	25																																				

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, cognitive level and course outcomes.				

Sign of the Course Coordinator

Signature of the HOD

Signature of the IQAC Chairman

9. CIE Practice Test model question paper

Program	Information Science and Engineering			Semester	4
Course Name	Web Development			Test	
Course Code	25IS44I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
Create a Currency Converter Design and build a simple web-based currency converter that allows users to: - Input an amount in one currency. - Select the currencies for conversion (e.g., USD to EUR). - Display the converted amount dynamically based on a fixed exchange rate. HTML Create a form with: - An input field for the amount. - Two dropdown menus to select the source currency and target currency. - A button to perform the conversion. - Add a section to display the converted amount. CSS - Style the form and result section for a professional look. - Add a consistent color scheme and spacing. - Use hover effects for the button and smooth transitions for dynamic updates. JavaScript Add functionality to: - Validate the input to ensure it is a positive number. - Use predefined exchange rates to calculate the conversion. For example: - USD to EUR: 1 USD = 0.85 EUR - EUR to USD: 1 EUR = 1.18 USD - USD to INR: 1 USD = 74.50 INR - Add more currencies as needed. dynamically display the converted amount.					50
Scheme of assessment Wireframing - 15 - Clarity and Structure - 05 - Consistency and Visual Appeal - 05 - Interactivity Representation - 05 Implementation – 30 - Use of correct HTML elements – 10 - CSS styling and responsiveness – 10 - JavaScript functionality – 10 Presentation -05					

Total Marks	50
-------------	----

Signature of the Course Coordinator

Signature of the HOD

10. SEE- Model Practice Question Paper

Program	Information Science and Engineering			Semester	
Course Name	Web Development			Test	II/IV
Course Code	25IS44I	Duration	180 min	Marks	50
Name of the Course Coordinator:					
Questions				CO	Marks
Design and build a Single Page Application (SPA) using React to manage a To-Do List. The application should allow users to add, edit, delete, and mark tasks as completed. The tasks should be stored temporarily (using local state or a temporary storage solution) and should be dynamically updated on the page without requiring a page reload. The application should have a clean, user-friendly interface and should demonstrate key features of React such as state management, event handling, and component reusability. Requirements - Allow users to add tasks with validation. - Enable task name editing. - Allow users to remove tasks from the list. - Toggle completion status. - Filter tasks by completion status. - Sort tasks by name or completion status. - Add and display due dates for tasks.					50
Scheme of assessment Wireframing and Design - 10 - Clarity and Structure - 05 - Consistency and Visual Appeal - 05 Implementation and Functionality– 30 - Event Handling - 10 - React Component Design and Creation- 10 - State Management and Routing – 10 Code Quality and Best Practices - 05 Presentation and Demo -05					
Total Marks					50

Signature of the Examiner

2) Signature of the Examiner

11 . Rubrics for Assessment of Activity(Qualitative Assessment)

Sl. No.	Dimension	Excellent	Good	Satisfactory	Needs Improvement (4 or below)	Score
		9-10	7-8	5-6	<=4	
1	Understanding of	Demonstrates thorough	Shows good	Demonstrates basic	Lacks understanding	8

	Concepts	understanding of underlying concepts and explains them confidently.	understanding of concepts with minor gaps.	understanding but struggles with deeper applications.	of concepts or is unable to explain them clearly.	
2	Implementation Skills	Writes efficient, well-structured, and commented code with appropriate logic.	Writes functional and mostly clean code with minor inefficiencies.	Writes basic code with some inefficiencies and lack of comments or structure.	Writes poor or non-functional code with little to no structure or comments.	6
3	Teamwork/Collaboration	Actively collaborates, contributes ideas, and supports peers effectively.	Participates well, shares ideas, and collaborates with minor lapses.	Participates minimally and may not engage fully with peers.	Does not participate in teamwork or creates disruptions within the team.	7
4	Presentation of Results	Explains the results and process clearly with accurate conclusions.	Presents results with minor inaccuracies or gaps in explanation.	Struggles to explain results clearly or makes incorrect conclusions.	Fails to present results clearly or shows lack of understanding of the process.	9
5	Adherence to Time	Completes the task well within the allotted time with no compromise on quality.	Completes the task on time with minor compromises in quality.	Completes the task with significant delay or compromises in quality.	Fails to complete the task within the allotted time or quality is highly compromised.	10
TOTAL						40

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

12. Equipment/software list with Specification for a batch of 30 students

Sl.No.	Particulars	Specification	Quantity
01	Computers	i3/i5	30
02	Internet Connection	100mbps	