Government of Karnataka
Department of Technical Education

Diploma Curriculum
Computer Science and Engineering

C 25 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 Computer 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/EC	25EE01I	Fundamentals of Electrical and 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	CS	25CS12T	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 1st semester to 6th semester(No Credits)											
Total				**16**	**0**	**16**	**32**	**24**	**250**	**-**	**100**	**-**	**100**	**-**	**450**

Note: The course 25EE01I shall be taught by faculty from the Electrical & Electronics (E&E) department. In the event that E&E faculty are not available in the institution, the course 25EE01I shall be assigned to faculty from the Electronics & Communication (E&C) department. If both E&E and E&C departments exist in the institution, the course 25EE01I shall be taught by the E&C faculty**.**

Government of Karnataka
DEPARTMENT OFTECHNICAL EDUCATION

Curriculum Structure

II Semester Scheme of Studies - Diploma in Computer 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	CS	25CS22T	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 1st semester to 6th semester (No Credits)											
Total				**17**	**0**	**16**	**33**	**25**	**250**	**-**	**100**	-	**100**	**-**	**450**

SEMESTER 1

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer Science and Engineering**	**Semester**	1/2
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:

In today's fast-changing digital world, foundational IT skills are crucial for technical professionals. This course equips students with hands-on experience in key areas, including computer fundamentals, cybersecurity, problem-solving, **Cloud Computing**, IoT, Artificial Intelligence (AI), and prompt engineering. Additionally, it covers IT certifications to help students build industry-relevant expertise and enhance their job readiness.

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

CO-01	Demonstrate knowledge of computer hardware, software, networking, and internet services.
CO-02	Identify common cyber threats and implement security measures.
CO-03	Apply algorithmic thinking and block-based coding to create simple programs.
CO-04	Explain applications of digital technologies such as Cloud, IoT and AI.
CO-05	Apply AI tools and prompt engineering techniques to generate meaningful outputs.

3. Course Content

Week	CO	PO	Lecture(3HRS) (Knowledge Criteria)	Practice(4HRS) (Performance Criteria)
1	1	1,4	**Introduction to Computers** ▪ Definition and basic understanding of a computer. ▪ Generations of Computers ▪ Classification of computer based on their size, purpose, functionality, and technology. ▪ Functional Block Diagram of a Digital Computer. **Memory Systems:** Types of Memory and Their Usage: Primary Memory, Secondary Memory: Input/output Systems	1. Identify the parts of a computer system. 2. Identify the operating system and hardware specifications of a computer.. 3. Basic folder/ file operations (GUI based) 4. Install application software such as web browser, scratch. 5. Hardware scavenger hunt (students identify components in disassembled PC images). Case Study: Prepare a report on important factors to be considered while buying a computer (based on purpose and budget).

			Software: System software vs. application software	
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 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. Explore and list 3 real-world examples for each type of network (LAN, MAN, WAN). 2. Find your Physical (MAC) and Logical (IP) Address 3. Create an email account (e.g., Gmail, Outlook) and explore its security settings 4. Using a Search Engine Effectively : Search for "How does a Search Engine work?" 5. Design, develop and host a personal website using any free platform such as wix.com or worldpress.com 6. Test Internet speed
3	**2**	**1,4, 7**	**Cybersecurity** ▪ Introduction to Cybersecurity ▪ What is Cybersecurity? ▪ CIA -triad ▪ Importance and Risks ▪ Common Threats: Malware, Phishing, Ransomware, Social Engineering ▪ Cybersecurity Best Practices **Secure Authentication and Access Control** ▪ Importance of Strong Passwords and ▪ Multi-Factor Authentication (MFA) ▪ Role of User Access Control and Privileged Accounts ▪ Password Management Tools **Safe Browsing and Data Protection** ▪ Secure Websites (HTTPS, SSL Certificates) ▪ Identifying Fake Websites and Links ▪ Basics of Encryption and Secure File Sharing ▪ Importance of Backups	1. Identify different cyber threats using real-world examples 2. Install and run an antivirus scan 3. Create strong passwords using password managers 4. Enable and test multi-factor authentication (MFA) 5. Implement User Access Control (UAC) settings on a system 6. Identify safe vs. unsafe websites using browser security indicators 7. Encrypt and decrypt a file using built-in OS tools 8. Set up and perform a basic data backup
4	**2**	**1,4, 5,7**	**Cyber security best practices** ▪ Awareness on cyber safety ▪ Do’s and dont’s w.r.t	1. Spot Fake Websites and Phishing Emails

			▪ Password Management ▪ Safe Browsing and Email Habits ▪ Software and System Security ▪ Data Protection and Backup ▪ Social Engineering and Phishing Awareness ▪ Secure Mobile and IoT Devices ▪ Staying Safe from Online Predators, Cyberbullying and Cyber harassment, Using Social Networks Safely.	2. Analyze real vs. fake websites (check for HTTPS, domain names, security certificates). 3. Identify phishing emails (hover over links, check sender email, grammar errors). 4. Update and Patch Management a. Check if your OS and software are up to date (Windows Update, Linux apt upgrade). b. Test an antivirus scan and remove unnecessary apps. 5. Implement a Backup Strategy 6. Encrypt and Secure Sensitive Files 7. Recognizing Scam Calls and Messages
5	**3**	**1,2,3,4,7**	**Introduction to Problem Solving** ▪ What is problem-solving? ▪ Problem-solving cycle. ▪ Introduction to block-based coding (Scratch, Blockly, MIT App Inventor / Klaritree or similar tool). ▪ Understanding algorithms, flowcharts, and sequencing.	1. Explore the interface of the block coding tool 2. Develop algorithms and draw flowchart ▪ for basic arithmetic operations. ▪ Metric conversions.
6	**3**	**1,2,3,4,7**	What are variables? Storing and updating values. Using variables for score counters and timers. Basic Elements of Block-Based Coding: ▪ Motion Blocks ▪ Looks Blocks ▪ Events Blocks ▪ Control Blocks ▪ Operators Blocks ▪ Variables Blocks	1. Create a simple animated sequence (e.g., making a sprite move in Scratch). 2. Design a flowchart for a real-world task 3. Create a score counter for a simple game. 4. Develop an interactive greeting app that responds to user input.
7	**3**	**1,2,3,4,7**	**Decision Making** ▪ What are conditions? (if, if-else, nested if). ▪ Boolean logic (AND, OR, NOT). ▪ Applying conditional logic in games and applications.	1. Develop algorithms and draw flowchart to demonstrate comparison and logical operations (eg. Comparison of two number) 2. Create an interactive story with decision-making (yes/no choices). **3.** Build a traffic light simulator using conditional statements.
8	**3**	**1,2,3,4,7**	**Understanding Loops and Repetition** ▪ Importance of loops in coding. ▪ Types of loops (repeat, repeat until, forever).	1. Create a bouncing ball animation using loops. **2.** - Design a counting program that prints numbers from 1 to 20 using loops.

			▪ Practical use of loops in problem-solving.	
9	**4**	**1,4, 7**	**Cloud Computing** ▪ What is **Cloud Computing**? ▪ **Cloud Computing** benefits and use cases ▪ Cloud service models (IaaS, PaaS, SaaS)	1. Create a free cloud account (AWS, Azure, or GCP) 2. Explore the cloud console and key services 3. Set up cloud storage and upload/download files 4. Create Online Forms and Surveys to capture data using cloud services
10	**4**	**1,4, 7**	**Internet of Things (IoT)** ▪ What is IoT? ▪ Characteristics ▪ Components of IoT (Sensors, Actuators, Communication, Cloud, Analytics) ▪ Use Cases of IoT across various industries. ▪ Examples of IoT in everyday life	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. 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	**4,5**	**1,4, 7**	**Artificial Intelligence** ▪ What is AI? ▪ Types of AI (Weak AI, Strong AI, General AI) ▪ AI in Everyday Life: Real-world applications ▪ AI Systems like prediction, recommendation ,etc. ▪ - AI Applications (Healthcare, Finance, Robotics, etc.)	Explore AI tools such as : ChatGPT, Deepseek, Gemini,Grok, Copilot, NapkinAI, Sora,etc
12	**5**	**1,4, 7**	**Prompt Engineering** ▪ What is Prompt Engineering? ▪ Role of AI language models ▪ Types of AI prompts: Direct, Instructional, Conversational ▪ Understanding AI capabilities and limitations **Structuring Effective Prompts** ▪ Key principles of writing effective prompts ▪ Clarity, specificity, and context in prompts ▪ Role of tone, format, and constraints	1. Exploring different AI models (ChatGPT, Claude, Bard) 2. Testing basic prompts and analyzing responses 3. Improving weak prompts 4. Experimenting with structured vs. unstructured prompts

			Domain-Specific Prompting ▪ Using AI for content creation (writing, marketing, coding) ▪ AI in education and research ▪ Customizing prompts for business applications	
13	**1,2,3,4**	**1,7**	**IT Certifications and Career Paths** ▪ Overview of IT certifications (Entry-level to Expert) ▪ Importance of certifications in IT careers ▪ Choosing the right certification based on career goals (Networking, Security, Cloud, Development, etc.)	**1.** Research and present a report on popular IT certifications. 2. Identify career interests and match them with relevant certifications. 3. Develop a career roadmap with certification milestones.

4. References

Sl No	Description
1	Computer Fundamentals by P.K. Sinha (11th Ed.)
2	Data Communications and Networking by Behrouz Forouzan (5th Ed.)
3	Cybersecurity for Beginners by Raef Meeuwisse – Covers threats
4	Coding for Kids: Scratch by Jon Woodcock
5	**Cloud Computing** Basics by Anders Lisdorf
6	IoT for Beginners by Adeel Javed
7	Artificial Intelligence: A Guide for Thinking Humans by Melanie Mitchell
8	The Tech Career Guide by Aki Ito
9	Learn Prompting
10	AI Playground
11	Tinkercad Circuits
12	Blockly Games
13	https://onlinecourses.swayam2.ac.in
14	https://www.geeksforgeeks.org
15	Essentials of Prompt Engineering | Coursera
16	https://www.ncerc.ac.in

5. Suggestive Online courses

Sl no	**Topic Name**	**Reference Courses**	**Self Assessment Link**	**Source**
1	**Cybersecurity**	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_014222737382490112870/overview		Coursera
2	**Security Attacks**	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0138424952317050881653l_shared/overview	https://infyspringboard.onwingspan.com/web/en/viewer/html/lex_auth_01384249741937049615982_shared?collectionId=lex_auth_01384249523170508816531_sharedandcollectionType=Course	IIHT
3	**Introduction to Problem Solving**	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0131149320724398081685_shared/overview	https://infyspringboard.onwingspan.com/web/en/viewer/iap/lex_auth_0132344659742228487432_shared?collectionId=lex_auth_0131149320724398081685_sharedandcollectionType=Course	Infosys Wingspan

4	**Flowcharts**	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0135015559136952327909/overview		Skillsoft
5	**Block coding**	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01317717283605708885_shared/overview	https://infyspringboard.onwingspan.com/web/en/viewer/html/lex_auth_013165205899452416510_shared?collectionId=lex_auth_01317717283605708885_sharedandcollectionType=Course	IIHT
6	Block coding	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0130944046684160001693_shared/overview		IIHT
7	**Cloud Computing**	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_29245015089922640000_shared/overview	https://infyspringboard.onwingspan.com/web/en/viewer/iap/lex_auth_01268242367501107260_shared?collectionId=lex_29245015089922640000_sharedandcollectionType=Course	Infosys Wingspan
8	Internet of Things	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_21553622882521997000_shared/overview	https://infyspringboard.onwingspan.com/web/en/viewer/iap/lex_12361814852557394000_shared?collectionId=lex_21553622882521997000_sharedandcollectionType=Course	Infosys Wingspan
9	Artificial Intelligence	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_8840337130015322000_shared/overview	https://infyspringboard.onwingspan.com/web/en/viewer/iap/lex_26105618936746710000_shared?collectionId=lex_8840337130015322000_sharedandcollectionType=Course	Infosys Wingspan

6. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1Theory Test	4	90	50	Average of all CIE=50 Marks
2.	CIE-2Practice Test	7	180	50	
3	CIE-3Theory Test	10	90	50	
4.	CIE-4Practice Test	13	180	50	
5	CIE-5 ▪ Portfolio evaluation (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
				Total	**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

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. Theory Test model question paper

Program	Computer 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. Explain the significance of the functional block diagram of a digital computer with a neat diagram. (5) b. Explain the evolution of computers through different generations, highlighting key technological advancements in each generation. (10) c. Explain the different types of networks (LAN, MAN, WAN) with suitable real-world examples. How do they differ in terms of scale and application?(10)	L2	1	25
2	a. Classify computers based on size and purpose. Provide one real-world use case for each type. (5) b. **Explain** how advancements in computer generations (from vacuum tubes to AI) have impacted business productivity. (10) c. Describe the client-server model using the example of an online banking website. (10)	L2	1	
	Section – 2			
3	a. A friend unknowingly clicks a phishing link and shares their bank credentials. Using the CIA triad, explain the potential risks. Then, outline steps they should take immediately to mitigate damage. (10) b. What is Multi-Factor Authentication (MFA)? How does it improve authentication security? Provide an example (8) c. Define Cybersecurity and explain the CIA Triad model. Why is it essential in today's digital world? (7)	L2	2	25
4	a. Describe the importance of password management tools. How do they contribute to secure	L2	2	

	authentication? Illustrate with examples of popular tools. (10) b. Explain how HTTPS and SSL certificates ensure secure browsing. How can users verify a website's security? (8) c. Compare phishing and ransomware attacks in terms of intent, delivery method, and impact. (7)			
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**

9. CIE Practice Test model question paper

Program	**Computer 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**
You have been hired as an IT Support Specialist in a company. Your first assignment is to set up a secure computer system, connect it to a network, and develop a simple automated task using block-based coding. You are required to: a. Set up a computer system by identifying its hardware and software specifications. ▪ Find and document system details like CPU, RAM, storage, and operating system. ▪ Identify whether the installed software is system software or application software. b. Configure network settings and verify the internet connection. ▪ Retrieve and document the IP address, MAC address, and default gateway. c. Ensure cybersecurity best practices to protect the system and online accounts. d. Develop a simple program using block-based coding (Scratch, Blockly, or MIT App Inventor) to automate a basic task. ▪ Create an interactive quiz that asks a user three questions and gives feedback on their answers. OR ▪ Develop a program where a sprite moves when arrow keys are pressed.	**2**	**50**
Scheme of assessment Computer System Setup - 10 Network Configuration and Internet Connectivity - 10 Cybersecurity Best Practices – 10 Block-Based Coding - 20		
Total Marks		**50**

Sign of the Course Coordinator **Signature of the HOD**

10. SEE- Model Practice Question Paper

Program	Computer Science and Engineering		Semester	1
Course Name	IT Skills	Course Code : 25CS01I	Duration	180 min
Questions			CO	Marks
As an IT specialist, you are tasked with setting up a secure digital environment for a small business. You must: a. Configure and document the computer hardware and software specifications of a system. b. identify the IP and MAC addresses and enable basic security settings. c. identify phishing threats and implement Multi-Factor Authentication (MFA). d. Develop a simple interactive program using block-based coding that automates a basic business task of greeting customers e. Use AI tools to generate a business report and refine the output using effective prompt engineering techniques. i.			1,2,3,5	50
Scheme of assessment a. System setup - 10 b. Cybersecurity Measures - 10 c. Block-Based Coding and Algorithmic Thinking - 10 d. Report Submission and Presentation - 20				
			Total Marks	**50**

1.Signature of the Examiner **2.Signature of the Examiner**

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

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer 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	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 **Binary Game (cisco.com)**

				Binary Bonanza! Binary Number game - Fun, Free, Online Way to Learn Binary (penjee.com)
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)	**Learn Binary Arithmetic - Binary Tutorial (ryanstutorials.net)** 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	**Every Circuit or similar simulator can be used** 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	**Every Circuit or similar simulators** 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. ***[Note : Concept and Application only]***	**Virtual Labs (vlabs.ac.in)** **simulator.io | Anonymous board** 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 Registers- definition and shift register types	simulator.io | Anonymous board . Examine the functionality of different flip flops **Simulation of**

			Counters: Asynchronous (Ripple) and Synchronous Counters. Compare combinational and sequential circuits ***[Note : Concept and Application only]***	. 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.	The central processing unit (CPU): Its components and functionality | Enable Sysadmin (redhat.com) 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	1,7	**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	1,7	**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	1,7	**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	1,7	**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	1,7	**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. References:

Sl No	Description
1	Digital Design – M. Morris Mano and Michael D. Ciletti
2	Computer Organization and Design: The Hardware/Software Interface – David A. Patterson and John L. Hennessy
3	Computer System Architecture – M. Morris Mano
4	Structured Computer Organization – Andrew S. Tanenbaum
5	Fundamentals of Logic Design – Charles H. Roth and Larry L. Kinney
6	Digital Circuits and Computer Organization - NPTEL (IITs and IISc) https://nptel.ac.in/courses/108/105/108105113/
7	MIT OpenCourseWare – Digital Systems https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/
8	Digital Design and Computer Architecture" (Harris and Harris)
9	Computer Organization and Design" (Patterson and Hennessy)
10	Logic and Computer Design Fundamentals" (Mano and Kime)
11	Logisim- Logisim for Beginners (YouTube).

5. CIE Assessment Methodologies

Sl.No	CIE Assessment	Test Week	Duration (minutes)	Max marks	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
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 (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
Total					**50 Marks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

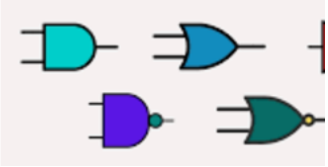
6. SEE - Theory Assessment Methodologies

Sl. No	SEE – Theory Assessment	Duration	Exam Paper Max marks	Exam Paper Max Marks scale down to (Conversion)	Min marks to pass
1.	Semester End Examination-Theory	3 Hours	100	50	20

7. CIE Theory Test model question paper

Program	**Computer 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)	L2	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 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)	L2	1	

	Section – 2			
3	a) Which gates would you use to design a circuit for these scenario, justify your selection (12M) a. The output should be 1 only when at least one input is 1. b. The output is 1 only when the two inputs are the same. c. In a control system, you need an output of 1 only when all inputs are 0. d. For a security system, if you want the output to be 1 only when both switche are in the off position. e. Output is 1 if and only if the inputs are different. f. 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)	L2 L3	2	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=11001and 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)	L2 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.				

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

8. CIE Practice Test model question paper

Program	**Computer Science and Engineering**			Semester	1
Course Name	Basics of Digital Logic and Computer Organization			Test	I/ III
Course Code	25CS11I	Duration	180 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. Equipment/software list with Specification for a batch of 30 students

Sl.No.	**Particulars**	**Specification**	**Quantity**
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1

06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2

SEMESTER 2

Government of Karnataka
DEPARTMENT OF TECHNICAL EDUCATION

Program	**Computer 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 programs using fundamental programming concepts.
CO-03	Design, code, and debug a Python program to solve a problem
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	For a given problem ▪ Identify the key components (input, output and logic) of an algorithm ▪ Develop a step-by-step algorithm to solve it.

			▪ Graph Problems ▪ String Processing Problems ▪ Numerical Problems ▪ Combinatorial Problems ▪ Cryptographic Problems **Algorithm** Representation: Natural Language, Pseudocode, Flowchart *[Note: Sorting Algorithms: designed to arrange data in a specific order* *Examples: Bubble Sort, Merge Sort, Quick Sort]*	
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**	**1,4**	**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**	**1,4**	**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**	**1,2, 4**	**Input and output statements:** input (); print (); Formatting output - string concatenation, format() and f-strings; Debugging techniques.	Practice Programs to understand reading input and formatting output
7	**3,4**	**1,2, 4**	Operators and their Precedence Expressions	Practice Programs to understand Operators and Expressions
8	**3,4**	**1,2, 3,4**	**Flow control** Conditional statements: if, if-else, if-elif, match case	Practice programs to understand the concept of conditional statements
9	**3,4**	**1,2, 3,4**	**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**	**1,2, 3,4,**	**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**	**1,2, 3,4**	**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**	**1,2, 3,4, 5,7**	**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**	**1,2, 3,4, 5,7**	**Modules and 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. References:

Sl No	Description
1	*Think Like a Programmer* (V. Anton Spraul)
2	*Automate the Boring Stuff with Python* – Al Sweigart
3	*Python Crash Course* – Eric Matthes
4	Introduction to Computation and Programming Using Python – John V. Guttag
5	How to Think Like a Computer Scientist: Learning with Python – Allen B. Downey
6	Programming for the Puzzled – Srini Devadas
7	Introduction to Algorithmic Thinking – Daniel Zingaro
8	CS50 Python Course - Harvard's CS50P: Introduction to Python
9	Google Python Course -
10	Codecademy Python Course
11	Coursera: Python for Everybody – University of Michigan

5. Suggestive Online courses

Sl no	Topic Name	Reference Courses	Self Assessment Link	Source
1	**Computational Thinking**	TOC - Problem Solving Using Computational Thinking | Infosys Springboard		Coursera
2	**Algorithmic thinking**	TOC - Programming Fundamentals using Python - Science Graduates - Foundation Program | Infosys Springboard		Infosys Wingspan
3	**Introduction to Programming**	TOC - Programming Fundamentals using Python - Science Graduates - Foundation Program | Infosys Springboard		Infosys Wingspan
4	**Basic Programming Concepts:**	TOC - Programming Fundamentals using Python - Science Graduates - Foundation Program | Infosys Springboard		Infosys Wingspan
5	**Introduction to python programming:**	TOC - Basics of Python | Infosys Springboard	Basics of Python - Self Assessment - Viewer Page | Infosys Springboard	Infosys Wingspan

6	**Input and output statements**:	[TOC - Programming Fundamentals using Python - Science Graduates - Foundation Program | Infosys Springboard]()	[Assessment - Programming Fundamentals using Python - Science Graduates - Viewer Page | Infosys Springboard]()	Infosys Wingspan
7	**Operators and their Precedence Expressions**			
8	**Flow control Practice programs to understand the Conditional statements**			
9	**Iterative statement**			
10	**Nested loops**			
11	**Functions**			
12	**Modules and Packages**			

6. Suggestive Program List

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	**Think Like a Programmer - Google Books** Think Like a Programmer: Introduction (youtube.com) General problem solving techniques- refer the book or youtube videos **https://www.codecademy.com/resources/blog/how-to-think-like-a-programmer/**
4	**Prepare Your Development Environment:** **1.** Download and install the necessary compiler or interpreter ▪ Python: Download from python.org. ▪ Java: Install the Java Development Kit (JDK) from Oracle or OpenJDK. 2. Verify the installation by checking the version using the terminal/command prompt 3. 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 = πr^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 ≥30°C, suggest "Wear light clothes." b. If 20°C ≤ temperature < 30°C, suggest "Wear moderate clothing." c. If temperature < 20°C, suggest "Wear warm clothes." 3. Implement a discount system for an e-commerce website. a. If the total purchase is ≥Rs5000, apply a 20% discount. b. If Rs2000 ≤ total < Rs5000, apply a 10% discount. c. If total < Rs2000, 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 **$S=1+2+3+\ldots\ldots+n$** **$S=1^2+2^2+3^2+\ldots\ldots+n^2$** **$H_n=1+1/2+1/3+\ldots\ldots+1/n$**
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 assign_priority() 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 calculate_interest() 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 apply_discount() 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.

7. CIE Assessment Methodologies

Sl.No	**CIE Assessment**	**Test Week**	**Duration (minutes)**	**Max marks**	
1.	CIE-1TheoryTest	4	90	50	Average of all CIE=50 Marks
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 (20) ▪ Online Course/s of minimum 10 Hrs. in Infosys Spring Board/ Swayam/NPTEL/AWS /any other (30)	1-13		50	
				Total	**50 arks**

Note:

Portfolio evaluation

Each laboratory exercise will be evaluated for a total of 20 marks. The evaluation will include the following components:

- Written description of the experiment in the observation book.
- The results obtained from the experiment.
- Corrections and evaluations of the experiment completed in the previous class, documented in the record book.

The average of all exercises shall be considered for the final assessment at the end of course.

Rubrics for the Mini Project (if included) should be defined by the course coordinator.

8. SEE - Theory Assessment Methodologies

Sl. No	**SEE – Theory Assessment**	**Duration**	**Exam Paper Max marks**	**Exam Paper Max Marks scale down to (Conversion)**	**Min marks to pass**
1.	Semester End Examination-Theory	3 Hours	100	50	20

9. CIE Theory Test model question paper

Program	**Computer 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)	L2,L3	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 algorithm to solve each. - (8M)	L2,L3	1	
	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.	L2	2	25

	e. What are syntax errors, logical errors, and runtime errors?			
4	a. How can learning programming enhance one's logical thinking, creativity, and problem-solving abilities? b. Explain the concept of "thinking like a programmer" and its importance in solving real-world problems. c. Explain how identifying patterns in problems can lead to efficient solutions. Use an example to demonstrate this process. d. Trace the evolution of programming paradigms through the generations of programming languages. e. 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.	L2	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**

10. CIE Practice Test model question paper

Program	**Computer 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**
Write a program that performs the following tasks: 1. Generate Random Numbers: Generate a random number within a given range [a, b], where **a** and **b** are user inputs. 2. Determine non-prime Number: Determine that generated random number is not prime. 3. 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.				**1,2,3,4**	**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**

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

Sl.No.	Particulars	Specification	Quantity
01	Desktop/Laptop PC with Windows/Linux	Intel i3, 500GB Hard Disk/SSD, 8GB RAM, Monitor, Mouse, Keyboard or higher configuration	30
02	Internet Connection	100 Mbps speed or higher subscription	1
03	LAN connectivity/ High speed Wireless AP	32 Port Switch with LAN cabling/ Wifi Adapters (32 No.)	1
04	Online UPS	5KV with 3 -6 hours backup	1
05	Projector	Multimedia Projector	1
06	White Board	Plane white board / Smart Board/Smart TV	1
07	Audio Speakers	Multimedia, Two-way hybrid speaker system	2