

First Year - Semester I and Semester II**Name of the Programme : Bachelor of Computer Applications****Course Code : CSA-100****Title of the Course : Problem-Solving and Programming****Number of Credits : 4 (3T + 1P)****Effective from AY : 2024-25**

Pre-requisites for the Course:	None	
Course Objectives:	1. To understand the concepts and techniques of problem-solving. 2. To analyze, understand, and build logic to solve basic problems. 3. To design Algorithms and flowcharts for better understanding and documentation for accurate implementation of the problem. 4. To code and implement a well-structured, robust programming logic using a suitable programming language.	
Units	Content	No of hours
I	Introduction to Problem Solving ● Problem-Solving Life Cycle — Understanding the Problem Statement, Analyzing the Problem, using Hierarchy charts, and Expressing Program logic using flowcharts / Pseudocode. ● Structured Programming concept ● Modular Programming-Top-Down Design, Bottom-Up Design, Stepwise Refinement Understanding basic Problem Solving Tools ● Algorithms: Definition and Attributes, Algorithm Constructs, Statements: Input-Output, Decision-making, and Looping, Examples ● Flowchart: Definition and its attributes, symbols, Statements: Input-Output, Decision-Making & Looping, Module representation, Drawing conventions and standards, Examples. ● Pseudo-code: Definition and its attributes, constructs, and Examples Basic Program Structures ● Data and its types (Integer, Floating-point, Character, String), Constants and variables, scope, instructions, and their types, how the computer stores data, Operators (Arithmetic, Assignment, Relational, Logical, etc), Expressions and Equations, Evaluation of expressions, and keywords. ● Local and Global Variables, Parameters, return values, naming conventions and standards, Understanding literals, syntax and semantics, functions, and modules.	15

II	Basic Sequential Instructions • Sequential statements using operators, constants, variables, operands, expressions, and equations. • Activity: Apply the concepts learn to design the algorithms of at least 2 basic problems. Represent it using flowchart and pseudo-code. Debugging & Documentation • Definition, Types, Need, and how to do it. Problem-solving with Decisions • The Decision Logic Structure, Multiple If/Then/Else Instructions, Using Straight-Through Logic, Using Positive and Negative Logic, Logic Conversion, Decision Tables, and Case Logic Structure. • Activity: Apply the concepts learned to design the algorithms for at least 4 basic problems. Represent it using flowchart and pseudo-code.	15
III	Problem Solving with Loops • The Loop Logic Structure, Incrementing, Accumulating, While/While End, Repeat/Until, Automatic-Counter Loop, Nested Loops, Indicators (flags). • Iterating, accessing, and modifying array elements. • Activity: Apply the concepts learnt to design the algorithms of at least 3 basic problems. Represent it using flowchart and pseudo-code. Problem Solving with Arrays • Arrays Concepts: One-dimensional Arrays, Creating, Concept of Strings, String as an array of characters. • Activity: Apply the concepts learnt to design the algorithms of at least 3 basic problems. Represent it using flowchart and pseudo-code. Understanding functions • Functions: Definition and its need and constructs, designing simpler functions, function communication using arguments, and return statements. scope of functions, function declaration and prototype, call by Value, and Call by reference. • Concept of Recursive functions: why, when, and how. Designing recursive functions and recursive calls. Basecase and recursive case. • Apply the concepts learnt to design the algorithms of at least three basic problems. Represent it using flowchart and pseudo-code.	15

IV	Practical work Using any suitable programming language like C, the concepts learned in the units from I to III are required to be implemented practically. The broad area of practical problems is mentioned/ suggested below.	30
Week 1 & 2 [These practicals should be done using pen, paper, and using buddy learning strategy]	1. For each of the following tasks, write a set of numbered, step-by-step instructions (a solution) so complete that another person can perform the task without asking questions. Define the knowledge base of this person by listing what you expect the person to know to follow your directions. For example, for task “a” (below), make a cup of cocoa, the knowledge base might include such things as knowledge of milk or water, a refrigerator, pan, spoon, cocoa, cup, range top or microwave, and so forth. - Make a cup of cocoa. - Sharpen a pencil. - Walk from the classroom to the student lounge, your dorm, or the cafeteria. - Start a car(include directions regarding what to do if the car doesn’t start). - Get a glass of water from your kitchen. - Start your computer. 2. Test your solution in problem 1 by giving your instructions to another person to see whether he or she can accomplish the task without your help. If they can’t, modify your solution so that the person can accomplish the task. Check the solution again by Giving the instructions to another person.	04
Week3 &4	3. Basic Program Structures - At least 10 basic programming problems related to Module II to be completed during the practical sessions. - More programs may be given to the learners to complete and practice as part of their Practice Work.	04
Week5 &6	4. Basic Sequential Instructions - At least 08 programming problems to be completed during the practical sessions. - More programs may be given to the learners to complete and practice as part of their Practice Work. 5. Debugging & Documentation - Debug & Document at-least 02 problems. - More programs may be given to the learners to complete and practice as part of their Practice Work.	04
Week7, 8&9	6. Problem Solving with Decisions - At least 08 programming problems to be completed during the practical sessions. - Debug & Document at least 02 problems. - More programs may be given to the learners to complete and practice as part of their Practice Work.	06

Week10 &11	6. Problem Solving with Loops ● At least 08 programming problems to be completed during the practical sessions. ● Debug & Document at least 02 problems. ● More programs may be given to the learners to complete and practice as part of their Practice Work.	04
Week12 &13	7. Understanding functions ● At least 08 programming problems to be completed during the practical sessions. ● Debug & Document at least 02 problems. ● More programs may be given to the learners to complete and practice as part of their Practice Work.	04
Week14 &15	8. Problem Solving with Arrays ● At least 08 programming problems to be completed during the practical sessions. ● Debug & Document at least 02 problems. ● More programs may be given to the learners to complete and practice as part of their Practice Work.	04
 Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem-Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real world when that's possible, it helps improve the student's understanding 7. To promote self-learning, give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	

References/ Readings:	Main Reading: 1. Forouzan, B.A., & Gilberg, R.F. (2007). <i>A Structured Programming Approach Using C</i>. Cengage Learning India. 2. Kuppuswamy, S., Malliga, S., Kanimozhi Selvi, C.S., & Kousalya, K. (2019). <i>Problem Solving and Programming</i>. Tata McGraw Hill. 3. Sprankle, M., & Hubbard, J.(2013). <i>Problem-solving and Programming Concepts</i>. Pearson Education India. Additional Reading: 1. K. N. King (2008). <i>C Programming: A Modern Approach</i>, 2nd Edition 2nd Edition, W. W. Norton & Company 2. Perry Greg, Miller Dean (2013). <i>C Programming Absolute Beginner's Guide</i> 3rd Edition, Kindle Edition. Que Publishing.
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basic concepts and terminologies of problem-solving, algorithms, flowcharts, pseudo-code, language syntax, and debugging. 2. Understand basic computing concepts, algorithm design, flowchart design, pseudo-code, programming constructs, and debugging. 3. Apply problem-solving and programming concepts and design solutions to simpler problems using algorithms, flowcharts, and pseudocode. 4. Code, debug, and analyze well-structured programming logic using suitable Programming language/s.

Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-111
Title of the Course : Computer System Fundamentals
Number of Credits : 4T
Effective from AY : 2024-25

Prerequisites for the Course:	Nil	
Course Objectives:	1. To remember the basics of computers, Computer Organization, Number Systems, process management, memory management, I/O Management, and File management concepts. 2. To understand the concepts of process management, memory systems, I/O devices, and File Management Systems 3. To apply the concepts of process management in handling deadlock situations. 4. To analyze the appropriate type of memory for a given scenario.	
Units	Content	No of hours
I	Fundamentals of Computer ● Evolution of Computer Operating Systems – Definition, Introduction to Major Functions/Services, OS Structure, Relationship between Kernel, OS, Hardware, Block Diagram of computer, Evolution of Computers - Computer Generations ● Computer Organization: Input Unit, Output Unit, Structure and functions of Central Processing Unit, Arithmetic Logic Unit, and Control Unit, Von Neumann Machine Architecture, Computer Function – Top Level View, Instruction Cycle with and without interrupts (State diagram), Classes of Interrupts, Multiple interrupts, Interconnection structures, Bus Interconnection. ● Number System Conversion(Binary, Decimal, Octal, Hexa-Decimal), Data Representation, Binary Arithmetic, One's and Two's Complement.	15
II	Processes & Process Management ● Process Definition, Process Control Block, Process States, Operations on Process. ● Threads Processes and Threads, Multithreading, Types of Threads. ● Process Scheduling Introduction, Scheduling Criteria, Scheduling Algorithms. ● Concurrency/Process Coordination Synchronization Principles, Mutual Exclusion, The Critical-Section Problem, Peterson's Solution ● Deadlock	15

	Principles, Deadlock Handling Methods, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	
III	Memory Management ● Memory Management Concepts Memory Partitioning (Fixed and dynamic), Swapping, Paging, and Segmentation. ● Virtual Memory Introduction, Demand Paging, Page Replacement- Algorithms, Thrashing. ● Cache Memory Characteristics of Memory Systems, Memory Hierarchy, Cache Memory Principles. ● Internal Memory Semiconductor main memory–SRAM, DRAM, Types of ROM. ● External Memory Magnetic Disk, SSD, Optical memory, Magnetic Tape	15
IV	Input/Output and File Management ● I/O Management I/O devices, Organization of I/O (programmed, interrupt driven and DMA), I/O Buffering, Disk Scheduling- Algorithms, RAID. ● File Management Overview–File and File Systems, File Structure, File Management System, File Organization and Access, File Directories, Directory Structure, File Sharing,	15
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT(Higher-Order Thinking) questions in class, which promotes critical thinking. 3. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 6. Discuss how every concept can be applied to the real	

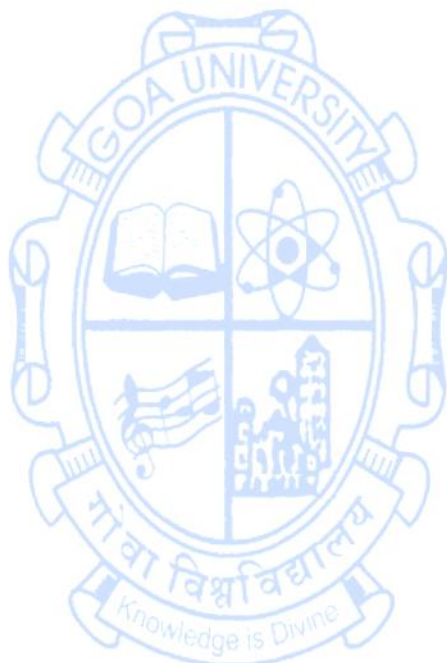
	world - and when that's possible, it helps improve the students' understanding 7. To promote self-learning, give at least one assignment where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.	
References/ Readings:	Main Reading: 1. Stallings, W.(2012). <i>Operating Systems: Internals and Design Principles</i>. Pearson Education. 2. Stallings, W.(2013). <i>Computer Organization and Architecture: Designing for Performance</i>. Pearson Education. Additional Reading: 1. Sinha, P. ,& Sinha, P.(2016). <i>Computer Fundamentals</i>. BPB Publications. 2. Silberschatz, A., Galvin, P.B., & Gagne, G. (2006). <i>Operating System Principles</i>. Wiley India.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the basics of computers, Computer Organization, Number Systems, process management, memory management, I/O Management, and File management concepts. 2. Understand the concepts of process management, memory systems, I/O devices, and File Management Systems 3. Apply the concepts of process management in handling deadlock situations. 4. Analyse an Appropriate type of memory for a given scenario.	

Name of the Programme : Bachelor of Computer Applications
Course code : CSA-112
Title of the Course : Open-Source Software
Number of Credits : 4T
Effective from AY :2024-25

Prerequisite for the Course:	None	
Course Objectives:	1. To remember the significance of Open-Source software practices and guidelines 2. To understand the Open-Source ecosystem, its use, impact, and importance. 3. To apply open-source methodologies, & case studies with real-life examples. 4. To collaborate and contribute to Open-Source Projects	
Unit	Content	No of Hour
I	Introduction to Open-Source Software • Open Source, Free Software, Free Software vs. Open-Source Software, Public Domain Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation, and the GNU Project. Methodologies • Open-Source History, Initiatives, Principles, and methodologies. Philosophy: Software Freedom, Open Source Development Model Licenses and Patents: What Is A License, Important FOSS Licenses (Apache, BSD, GPL, LGPL), copyrights and copy lefts, Patents Economics of FOSS: Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization.	15
II	Social Impact • Open source vs. closed source, Open-source government, Open-source ethics. Social and financial impacts of open-source technology, Shared software, Shared source, and Open Source in Government. Case studies • Example Projects: Apache web server, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, WordPress, GCC, GDB, GitHub, Open Office. Study: Understanding the developmental models, licensing, mode of funding, and commercial/non-commercial use. Opensource Hardware, Open-Source Design, Open-Source Teaching. Open-source media.	15

III	Collaboration, Community, and Communication Contributing to Open-Source Projects ● Introduction to Git Hub, interacting with the community on Git Hub, Communication and etiquette, testing open-source code, reporting issues, and contributing code. ● Introduction to Wikipedia, contributing to Wikipedia, or contributing to any prominent open-source project of the student's choice. ● Starting and Maintaining own Open-Source Project	15
IV	Understanding Open-Source Ecosystem ● Open-Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open-source hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, debuggers, Programming languages, LAMP, Open Source database technologies	15
Pedagogy:	1. Course delivery pattern, evaluation scheme, and prerequisite shall be discussed at the beginning. 2. Lectures preferably to be conducted with the aid of a multi-media projector, blackboard, group activities, charts, cases, etc. 3. One internal written exam would be conducted as a part of the internal theory evaluation. 4. One assignment based on the course content may be given to the students to evaluate how the learning of objectives was achieved.	
References:	Main Reading: 1. Fogel, K. (2009). <i>The Open Source Way: Openness and Collaboration Principles for Life</i>. O'Reilly Media. 2. Fogel, K. (2005). <i>Producing Open Source Software: How to Run a Successful Free Software Project</i>. O'Reilly Media. 3. Hassan, N. A. (2018). <i>Open Source Intelligence Methods and Tools: A Practical Guide to Online Intelligence</i>. Apress. 4. Raymond, E. S. (1999). <i>The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary</i>. O'Reilly Media. Additional Reading: 1. Das, S. (2017). <i>UNIX: Concepts and Applications</i>. Tata McGraw Hill Education. 2. DiBona, C., Cooper, D., & Stone, M. (Eds.). (2005). <i>Open Sources 2.0: The Continuing Evolution</i>. O'Reilly Media. 3. Helmke, M., Joseph, E.K., Rey, J.A., Ballew, P., & Hill, B.M. (2014). <i>The Official Ubuntu Book</i>. Prentice Hall. 4. Whitehurst, J. (2015). <i>The Open Organization: Igniting Passion and Performance</i>. Harvard Business Review Press.	
Course Outcomes:	On completion of the course, students will be able to: 1. Remember the significance of Open-Source software practices and guidelines. 2. Understand the Open-Source ecosystem, its use, impact, and	

	importance. 3. Apply Open-Source methodologies, and case studies with real-life examples. 4. Collaborate and contribute to Open-Source Projects
--	--



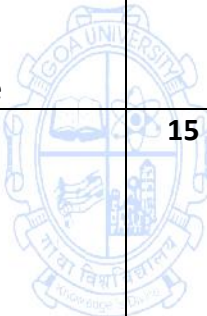
Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-131
Title of the Course : E-Commerce
Number of Credits : 3T
Effective from AY : 2024-25

Prerequisites for the Course:	None	
Course Objectives:	1. To give a fundamental understanding of e-commerce and online marketing 2. To instill ideas of Search Engine Optimization and Marketing, Applications of e-commerce and digital payments 3. To identify, define and differentiate the e-commerce models and risks of electronic commerce.	
Units	Content	No. of Hours
I	Introduction to Electronic Commerce: Meaning, Nature, and scope of e-commerce, History of e-commerce, Business applications of e-commerce, E-Commerce Models(B2B, B2C, C2C, B2G), Advantages and Disadvantages of e-commerce, Applications of M-Commerce. E-Commerce Web-sites: Websites as a marketplace, Role of the website in B2C e-commerce, Website design principles, Alternative methods of customer communication such as e-mail, Email etiquette, and e-mail security. Online Marketing: Online marketing and advertising, Push and pull approaches, Web counters, Web advertisements, Content marketing, Need of Digital Marketing for an e-commerce Business.	15
II	Search Engine Optimization: Search Engine Optimization (SEO), Search Engine Marketing (SEM), Social Media Marketing (SMM), Web Analytics. Applications of E-commerce: Applications of e-commerce to Supply chain management Applications of e-commerce to Customer Relationship Management, Product and service digitization, Remote servicing. Electronic Payment System: Types of payment systems, credit cards, debit cards, mobile, etc., Electronic Fund Transfer (EFT), Operational credit and legal risk of e-payment, and Risk management options for e-payment systems.	15
III	Business to Consumer E-Commerce: Cataloguing, Order planning and order generation, Cost estimation and pricing, Order receipt and accounting, Order selection and prioritization, Order scheduling, Order fulfilling, Order delivery, Order billing, Post sales service.	15

	Business-to-Business E-Commerce: Need and Models of B2B e-commerce, Using public and private computer networks for B2B trading; EDI and paperless trading, Characteristic features of EDI service arrangement, EDI architecture, and standards. Security Issues in E-Commerce: Risks of e-commerce, Types and sources of threats; Security tools, Risk management approaches.
Pedagogy:	PowerPoint, Tutorials, Hybrid learning.
References/Readings:	Main Reading: 1. Kalakota, Ravi, Andrew Whinston(2015). Frontiers of Electronic Commerce. Pearson Education. 2. P.T.Joseph(2015).E-Commerce: An Indian Perspective Paperback. PHI Learning. 3. V.Rajaraman(2015). Essentials of E-Commerce Technology. PHI Learning. Additional Reading: 1. C.S.V.Murthy (2015). <i>E-Commerce - Concepts, Models and Strategies</i>. Himalaya Publishing House.
Course Outcomes:	At the end of the course, students will be able to: 1. Understand the foundation of e-commerce, e-commerce websites and Online Marketing and Security Issues 2. Explaintheimportanceof Search Engine Optimization, Applications of E-commerce and Electronic Payment Systems. 3. Compare B2B and B2C e-commerce models.

Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-132
Title of the Course : Green Computing
Number of Credits : 3T
Effective from AY :2024-25

Pre-requisites for the Course:	None	
Course Objectives:	1. To remember the fundamentals of Green Computing and Green IT 2. To understand Green Hardware/Software and green Data Centers. 3. To devise a Green IT Strategy for an organization. 4. To implement Green IT initiatives.	
Units	Content	No of hours
I	Trends and Reasons to Go Green ● Overview and Issues ● Current Initiatives and Standards ● Consumption Issues-Minimizing Power Usage, Cooling Introduction to Green IT ● Green IT ● Holistic Approach to Greening IT ● Awareness to Implementation ● Green IT Trends ● Green Engineering Greening by IT ● Using RFID for Environmental ● Sustainability ● Smart Grids ● Smart Buildings and Homes ● Green Supply Chain and Logistics ● Enterprise-Wide Environmental Sustainability Green Hardware and Software ● GreenHardware ● Introduction ● Life Cycle of a Device or Hardware ● Reuse, Recycle, and Dispose ● Green Software ● Introduction ● Energy-Saving Software Techniques	15
II	Green Data Centres and Storage Green Data Centres ● Data Centre IT Infrastructure ● Data Centre Facility Infrastructure: Implications	15

	for energy efficiency • IT Infrastructure Management • Green Data Centre Metrics Green Data Storage • Introduction • Storage Media Power Characteristics • Energy Management Techniques for Hard Disks • System-Level Energy Management Green Networks and Communications • Introduction • Objectives of Green Network Protocols • Green Network Protocols and Standards Enterprise Green IT Strategy • Introduction • Approaching Green IT strategies • Business Drivers of Green IT Strategy • Business Dimensions for Green IT Transformation • Organizational Considerations in a Green IT Strategy • Steps in Developing a Green IT Strategy • Metrics and Measurements in Green Strategies • Organizational and Enterprise Greening • Greening the Enterprise: IT Usage and Hardware	
III 	Managing and Regulating Green IT Managing Green IT • Introduction and Approaches to Green • Strategizing Green Initiatives • Implementation of Green IT • Information Assurance • Communication and social media Regulating Green IT • Introduction • The Regulatory Environment and IT Manufacturers • Non-regulatory Government Initiatives • Industry Associations and Standards Bodies • Green Building Standards • Green Data Centres • Social Movements and Greenpeace The Future of Green IT • Green Computing and the Future • Mega trends for Green Computing • Tele-presence Instead of Travel • Tele-commuting instead of Commuting	15 

Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning, etc. 2. Ask at least three HOT(Higher-Order Thinking)questions in the class, which promotes critical thinking. 3. Adopt Problem-Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with creative ways to solve them. 6. Discuss how every concept can be applied to the real world and when that's possible, it helps improve the students' understanding 7. To promote self-learning give at least one assignment (equivalent to 50% assignment weightage) where they can complete one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
References/ Readings:	Main Reading: 1. San Murugesan, G.R.Gangadharan(2013).<i>Harnessing Green IT: Principles and Practices</i>. Wiley. 2. Toby Velte, Anthony Velte (2008). <i>Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line</i>. McGrawHillEducation. Additional Reading: 1. Bud E. Smith (2013). Green Computing- Tools and Techniques for saving energy, money and resources. Auerbach Publications. 2. MarkG. O'Neill (2011) Green IT for Sustainable Business Practice. BCS, The Chartered Institute for IT. 3. Mike Ebbers, Alvin Galea (2008). The Green Data Center: Steps for the Journey. International Business Machines Corporation 2008.
Course Outcomes:	On completion of the course, students will be able to: 1. Recall the fundamental concepts of Green Computing and Green IT 2. UnderstandfundamentalsofGreenComputingandGreenITanditsregulation. 3. Apply Green IT Strategies for an organization. 4. Analyze Green IT/Computing regulation and strategies.

Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-141
Title of the Course : Office Automation and PC Troubleshooting
Number of Credits : 3 (1T + 2P)
Effective from AY :2024-25

Pre-requisites for the course:	Nil	
Course Objectives:	• To understand the basics of office automation software and its applications. • To develop proficiency in using word processing, spreadsheet, and presentation software. • To diagnose and troubleshoot common PC issues and optimizing the performance of a PC.	
Units	Content	No of hours
I	Introduction to Office Automation Understanding office automation software and its applications, Types of office automation software, Microsoft Office Suite, Google Workspace	15
	Word Processing Introduction to Microsoft Word, creating and formatting documents, working with templates, Mail merge and labels, Collaboration tools	
	Spreadsheets Introduction to Microsoft Excel, creating and formatting spreadsheets, working with formulas and functions, Charts and graphs, Collaboration tools	
	Presentation Software Introduction to Microsoft PowerPoint, creating and formatting presentations, working with images, videos, and animations, Collaboration tools	
	Email management & Internet and Web Browsers Introduction to Email, setting up and configuring email accounts, composing and sending emails, Managing Email Accounts Introduction to the Internet, Web browsers, searching the Internet, configuring web browser settings PC Troubleshooting Hardware Troubleshooting: Basic hardware components of a PC, Common hardware issues and their solutions, maintenance, and optimization of hardware Software Troubleshooting: Common software issues and their solutions, Malware and virus removal, System recovery and backups, Network Troubleshooting	
II	Practical: list of suggested practical's	60

Week 1&2	1. Study of Google Workspace and its collaboration tools ● Create a Google form to build a questionnaire and collect responses. ● Use the tool to take surveys and generate reports on them.	08
Week 3&4	2. Experiments based on Word processing ● To create a document and apply basic formatting, creating a bulleted and numbered list, applying headers and footers to the document, and page numbering. ● To study the creation of tables in MS Word and apply formatting to the table ● To insert pictures, shapes, and clipart in a document ● Prepare a bio-data in MS word using templates.	08
Week 5	3. Experiments based on Mail Merge ● Using Mail Merge to prepare letters, email messages, envelopes, and labels. ● Prepare ease-to-field trip notices using mail merge	04
Week 6to8	4. Practical on Spreadsheet ● Create a worksheet and perform basic formatting of cells, rows, and columns. ● Create a Student Mark Statement in MS Excel and calculate total, average, and percentage using Auto sum. ● Apply conditional formatting to the mark statement. ● Working with an advanced formulae ● Presenting data with charts	10
Week 8 to10	5. Practical Presentation software ● Usage of text, images, and animation for presentation ● Adding slide transition, custom animation, and setup show. ● Creating graphs in presentation. ● Design an advertisement in MS PowerPoint	10
Week 11	6. Email Management ● Experiment to setup and configure the email account ● Compose and send an email to at least 5 email addresses ● To manage the Email Accounts	08
Week 12 &13	7. Practical Internet browsing, downloading files, knowing secure browsing.	04
Week 14 &15	8. PC troubleshooting ● Understanding PC components and PC assembling, formatting, fragmentation and installation of Operating systems and configuration of different types of software. ● To install different hardware devices, configure printers ● Identifying issues with hardware devices	08

	and troubleshooting. • Network setup of two or more PCs. • To install an antivirus software and understand the working of the firewall	
Pedagogy:	Suggested strategies to use to accelerate the attainment of the various course outcomes. 1. Lecture methods need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning etc. 2. Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes creative thinking. 3. Adopt Problem Based Learning(PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with creative ways to solve them. 6. Discuss how every concept can be applied to the real world and when that's possible, it helps improve the students' understanding 7. To promote self-learning, give at least one assignment where they can complete atleast one MOOCs(certificator equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 8. Activity/ Practical Based Learning (Suggested Activities in Class) a. Real-world problem solving using group discussion. E.g., designing posters for road safety etc., b. Demonstration of solution to a problem through design. 9. Demonstration of simple projects and motivating the students to develop similar type of projects.	
References/ Readings:	1. Andrews,J. (2019).A+ Guide to IT Technical Support (MindTap Course List). Cengage Learning. 2. Shelly,G.B.,&Vermaat,M.E.(2017).MicrosoftOffice365&Office2016 :Introductory.Cengage Learning 3. Vermaat,M.E.(2022).DiscoveringComputer: Digital Technology, Data, and Devices. Course Technology Inc.	
Course Outcomes:	On completion of the course, students will be able to: 1. Understand the basics of office automation software 2. Demonstrate proficiency in creating and formatting documents, spreadsheets, and presentation 3. Analyze the basic software and hardware issues & troubleshoot them.	

Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-142
Title of the course : Python Programming
Number of Credits : 3 (1T +2P)
Effective from AY :2024- 25

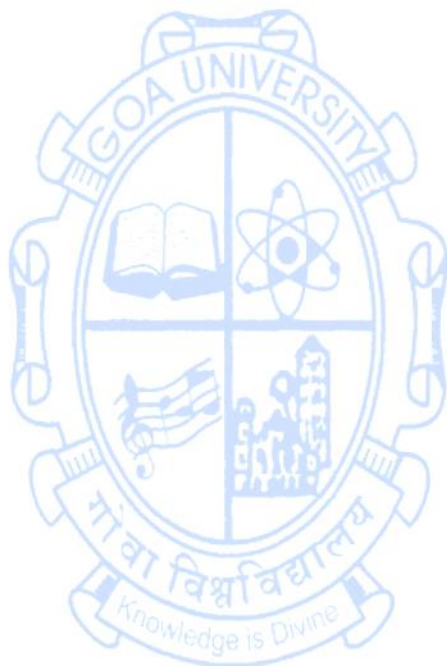
Prerequisite for the course:	None	
Course Objectives:	1. To understand Python programming concepts. 2. To acquire proficiency in utilizing Python library functions and data structures. 3. To gain fundamental understanding of object-oriented programming (OOPS) concepts in Python.	
Units	Content	No of Hours
I	Introduction to Python - Python interpreter/shell, indentation; identifiers and keywords; literals, numbers, and strings; operators(arithmetic operator, relational operator, Boolean operator, assignment, operator, ternary operator and bitwise operator) and expressions. Program Flow Control - Input and output statements, defining functions, control statements (conditional statements, loop control statements, break, continue and pass, exit function.), default arguments, errors, and exceptions. List,Tuple and Dictionary - Lists creation, traversal, slicing and splitting operations, passing list to a function. Tuple and Dictionaries. OOPS Concepts - Introduction to Classes, Objects and Methods, Standard Libraries, File handling through libraries.	15
II	Practical Work -I Using any suitable pythonIDE or Interpreter.	Practical Hours(28)
Week1	1. Write a Python program to find the area and perimeter of a circle. 2. Write a Python program to generate the Fibonacci series. 3. Write a Python program to compute the GCD of two numbers. 4. Write a Python program to generate the first prime numbers. 5. Write a Python program to find the sum of squares of n natural numbers.	4
Week2 & week3	6. Program palindrome or not 7. Write a Python program to store strings in a list and	6

	print them. 8. Write a Python program to find the length of a list, reverse it, copy it, and then clear it. 9. Write a Python program to print the squares of numbers from 1 to 10 using loop control. 10. Write a Python program to count the number of even and odd numbers from a series of numbers. Sample numbers: numbers= (1,2,3,4,5,6,7,8,9) Expected Output: Numberofevennumbers:5Numberofoddnumbers:4	
Week4 & week5	11. Write a Python program that prints all the numbers from 0 to 6 except 3 and 6 Note: Use the 'continue' statement. Expected Output: 0 1 2 4 5 12. Print the following pattern 1 12 123 1234 12345 13. Display numbers from -10to-1 using for loop 14. Print the following pattern <pre> * * * * * * * * * * * * * * *</pre> 15. Write a Python function to sum all the numbers in a list Sample List: (8, 2, 3, 0, 7) ExpectedOutput:20	8
Week6 & week7	16. Write a Python program to reverse a string Sample String: "1234abcd" ExpectedOutput:"dcba4321" 17. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument 18. Write a Python program to print the even numbers from a given list. Sample List:[1,2,3,4,5,6,7,8,9] Expected Result:[2,4, 6,8] 19. Write a Python program to calculate the length of a string 20. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '\$', except the first char itself.	10
III	Practical Work -II	Practical Hours(32)

Week8 & week9	21. Write a Python program to change a given string to a new string where the first and last chars have been exchanged. 22. Write a Python program to count the occurrences of each word in a given sentence 23. Write a program to find the first and the last occurrence of the letter 'E' and character ',' in "NEP IMPLEMENTATION, FOR BCA ". 24. Write a program to check if the word 'open' is present in the "This is open-source software". Write a program to check if the letter 'e' is present in the word 'Welcome'.	10
Week10 & week11	26. Write a program in Python to delete first and last elements from a list 27. Write a Python program to check a list is empty or not 28. Write a Python program to remove duplicates from a list 29. Write a Python program to find the second smallest number in a list 30. Write a Python program to find common items from two lists 31. Let list=['a','b','c','d','e','f']. Find a)list[1:3]b)t[:4]c)t[3:] 32. Write a Python program to create a tuple with different data types.	6
Week12 & week13	33. Write a Python program to unpack a tuple in several variables 34. Write a Python program to read an entire text file 35. Write a Python program to append text to a file and display the text 36. Write a Python program to count the number of lines in a text file 37. Write a Python program to write a list to a file 38. Write a Python program to extract characters from various text files and puts them into a list	6
Week14 & week15	39. Write a function that reads a file file1 and copies only alternative lines to another file file2. Alternative lines copied should be the odd-numbered lines. 40. Write a function that reads a file file 1 and displays the number of words and the number of vowels in the file. 41. Consider a showroom of electronic products, where there are various salesmen. Each salesman is given a commission of 5%, depending on the sales made per month. In case the sale done is less than 50000, then the salesman is not given any commission. Write a function to calculate total sales of a salesman in a month, commission and remarks for the salesman. Sales done	10

	by each salesman per week is to be provided as input. Assign remarks according to the following criteria: Excellent: Sales ≥ 80000 Good: Sales ≥ 60000 and < 80000 Average: Sales ≥ 40000 and < 60000 Work Hard: Sales < 40000	
Pedagogy:	Suggested strategies to use to accelerate the attainment of the various course outcomes: - Lecture methods need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use - Video/Animation to explain various concepts. - Collaborative, Peer, Flipped Learning etc. - Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking. - Adopt Problem-Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. - Introduce Topics in manifold representations. - Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. - Discuss how every concept can be applied to the real-world and when that's possible, it helps improve the students' understanding. To promote self-learning give at least one assignment where they can complete at least one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. - One internal practical exam will be conducted as a part of internal evaluation. - Practical shall be performed in the laboratory as indicated in the syllabus. - A Handwritten Hard Copy of journal shall be maintained clearly mentioning the name of the experiment and other required information.	
References:	Main Reading: - Balagurusamy, E. (2017). <i>Introduction to Problem Solving with Python</i>. McGraw Hill Education India Private Limited. - Nageshwara Rao, R. (2018). <i>Core Python Programming</i>. Dreamtech Press. - Sedgewick, R., Wayne, K., & Dondero, R. (2016). <i>Introduction to Programming in Python: An Interdisciplinary Approach</i>. Pearson India Education Services Pvt. Ltd. - Yates, J. (2019). <i>Python Practical Python Programming For Beginners and Experts</i>. Packt Publishing. Additional Books - Dawson, M. (2020). <i>Python Programming for the Absolute Beginner</i>. No Starch Press. - Kumar, T. (2018). <i>Python Programming</i>. Wiley.. - Hoskins, A. (2017). <i>The Python Book: The ultimate guide to coding with Python</i>. Future Publishing Limited..	

	4. Shovik, J. (2019). <i>Python All-In-One for Dummies</i> . For Dummies.
Course Outcomes	On completion of the course, students will be able to: 1. Remember the basics of Python Programming 2. Understand the concepts and constructs of Python programming. 3. Apply Python library functions and data structures. 4. Analyze the implementation of Python Programming

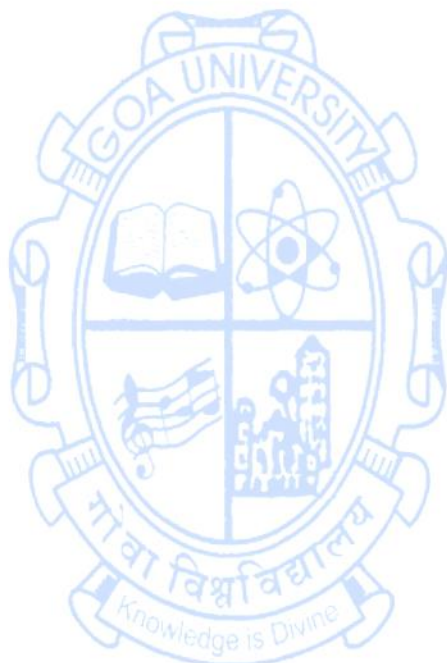


Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-143
Title of the Course : Data Analytics using Spreadsheets
Number of Credits : 3P (1T +2P)
Effective from AY : 2024-25

Pre-requisites for the Course:	Nil	
Course Objectives:	1. Remember basic and advanced functions in spreadsheets. 2. Understand data analysis and data visualization with charts and pivot tables. 3. Implement dataanalysis tools and functions.	
Unit	Content	No of Hours
I	Introduction to spreadsheets Introduction to spreadsheets, understanding spreadsheet environment, cell addressing, cell references, absolute and relative cell references, named ranges, formatting using paste special, Data filters and sorting, worksheet and workbook protection	15
	Formulas and Functions, Advanced Functions - Sum, Average, Min, Max, count, IF, nested IF, using IF with AND OR formulas, COUNTIF, SUMIF, AVERAGEIF formulas, TEXT functions - Vlookup function, match function, index function, date and time functions, maths functions, financial functions	
	Data Analysis, Charts and Visualization - Conditional formatting, What if analysis using data table, Goal seek, scenario manager, Linear regression - Data storytelling tips, Introduction to charts, types of charts, uses and benefits, - Understanding Pivot tables, Pivot table tips and tricks	
	DAX and Power Query Power query tips, Introduction to power pivot, Apply DAX in power pivot for analysis, introduction to types of joins in power query, full outerjoin and innerjoin in powerquery, left outer join and right outer join in power query, Left antijoin and right antijoin in power query	
	Dashboard reporting and Data Analysis tools - Understanding how to create a dashboard in spreadsheets, a Sales Analytical Dashboard using Data Analysis Expressions (DAX) & Visualization, creating a simplified GANTT chart with AND function - ANOVA, Correlation, Covariance, regression, sampling, t-test, z-test and histograms	
	PRACTICALS	60 hours
UNIT II	List of suggested practicals	28

Week1	Practical on introduction to a spreadsheet using simple tabular data and formatting using paste special, absolute, and relative cell references, calculating sum, average, min, max, count, and percentage.	4
Week2	Practical using IF, NESTEDIF, SUMFIF, AVERAGEIF, COUNTIF	4
Week 3 &4	Practical on advanced functions	8
Week5	Practical on conditional formatting, what-if analysis using Goal seek, scenario manager and linear regression	4
Week 6 &7	Practical on different types of charts and pivot table with suitable examples	8
UNIT III	List of suggested practicals:	32
Week8 to10	Practical on Powerquery, DAX, and different types of joins with suitable data.	12
Week 11 & 12	Creating dashboard and gantt chart in spreadsheet using suitable examples	8
Week13to15	Excel data analysis Toolpak add-in covering ANOVA, Correlation, Covariance, Descriptive Statistical analysis, random number generation analysis, rank and percentile analysis, regression analysis, T-test, Z-test, Histogram	12
Pedagogy	Suggested strategies to use to accelerate the attainment of the various course outcomes. - The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use - Video/Animation to explain various concepts. - Collaborative, Peer, Flipped Learning etc. - Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking. - Adopt Problem-Based Learning(PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. - Introduce Topics in manifold representations. - Show the different ways to solve the same problem and encourage the students to come up with creative ways to solve them. - Discuss how every concept can be applied to the real world-and when that's possible, it helps improve the students' understanding - To promote self-learning give at least one assignment where they can complete at least one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. - One assignment in the form of a mini-project collecting data and using analytic tools may be given to the students.	
References	Main Reading: D.Whigham(2007).Business Data Analysis using Excel. NewYork: Oxford University Press.	

	2. Michael Alexander, Richard Kusleika, John Walkenbach. (2018). Excel 2019 Bible Paperback. Wiley 3. StephenL. Nelson, Elizabeth C.Nelson,(January 2018).Microsoft Excel Data Analysis for Dummies. Wiley. 3ed
Course Outcomes	1. Demonstrate basic and advanced functions in spreadsheet applications. 2. Apply data analysis techniques and create visualizations using charts and pivot tables. 3. Implement data analysis tools and functions for practical applications.



Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-144
Title of the Course : 2DAnimation
Number of Credits : 3 (1T + 2P)
Effective from AY : 2024-25

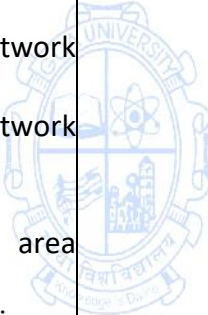
Pre-requisites for the Course:	None	
Course Objectives:	1. Familiarize with various approaches, methods and techniques of Animation 2. Study the Basics of Color Theory and Graphics. 3. Implement traditional & digital tools to produce still and moving images. 4. Develop expertise in life drawing and related techniques.	
Units	Content	No. of hours
I	Introduction to Animation: • Terms used in Animation, • Types of Animation-Cel(Celluloid) Animation, 2D Animation, 3D Animation, Motion Graphics, Stop Motion. • Animation Techniques used in 2D Animation: Hand-drawn animation, Cut-out animation, Model animation or Stop motion animation, Computer animation, or computer-generated imagery. • Equipment required for animation-Pentablet, Graphic tablet, Artist glove, Ergo stand, Flex arm. Principles of Animation: • Disney's twelve basic principles of animation- Squash and stretch, Anticipation, Staging, Straight ahead action and pose to pose, Follow through and overlapping action, Slow in and slow out, Arc, Secondary action, Timing, Exaggeration, Solid drawing, Appeal Fundamentals of Drawing and Design • Basic Shapes and Drawing techniques • Concepts of Visualization-Perspective drawing, Illustration, Shading, and Sketching techniques Color Theory and Graphics • Color fundamentals-primary colors, secondary colors, Tertiary Colors • Properties of color-Hue, Reflective Value, Tints, And Shades, Saturation, Color tone – Intensity • Additive Color System (RGB)-Subtractive Color System (CMYK). • Vector and Raster graphics	15

	2D Animation tools processing 2D animation software paradigms- scripting & Story boarding, Usage of tools for Digital Painting and vector drawings, developing a character and background creation.	
II	Practical Work Suggested list of Animation Tools: Pencil2d, Adobe Animate, Synfig studio, OpenToonz	Practical Hours (60)
Week1	Flipbook(on paper) Drawing simple flipbook with minimum 10 pages Flip Book (Digital) Create simple flipbook with minimum 10 frames	(4)
Week 2&3	Frame by frame animation Creating simple frame by frame animation for a short animation, demonstrating the concept of layering and onion skinning (maximum 20sec with color drawings and background.)	(8)
Week 4&5	Tween Create simple animations, using concepts of Grouping layers to create artwork, import images and apply tweening, Preview, and Render the animation in suitable format ● a classic Tweening: Create an E-card animation ● Motion tweening: Creating animation: Draw, Give Rotation effect, Time Loop demonstration ● Shape tweening: Demonstrate the animation	(8)
Week 6&7	Ball animation Drawing the ball with gradient color, Creating key frames for the animation sequence, Creating stretch and squash for the ball animation, Giving tween to the sequence of ball animation by connecting to path, duplicating waypoints, work with background image in the developed scene	(8)
Week 8&9	Character Animation Drawing simple character, Preparing the character for animation, dividing each body parts into symbol and creating motion	(8)
Week 10to12	Human/Animal walk cycle Drawing cycle sheet for an human/animal walk cycle, Creating four different types of walk cycle(jump,run,tip toe, crawl)	(12)
Week 13to15	Mini project Prepare a storyboard and create short animation film using the concepts learnt in previous weeks	(12)

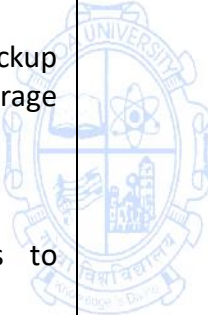
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. The lecture method need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. You may use a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning etc. 2. Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking. 3. Adopt Problem-Based Learning(PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 4. Introduce Topics in manifold representations. 5. Show the different ways to solve the same problem and encourage the students to come up with creative ways to solve them. 6. To promote self-learning, give at least one assignment where they can complete at least one MOOCs(certificate or equivalent)course out of lecture hour. Test their understanding through quizzes or presentations. 7. Mini-Project may be given as a part of the assessment
References/Readings:	Main Reading: 1. Chris Patmore(2003).The Complete Animation course. Barrons Educational Series. 2. Mary Murphy(2008). Beginner's Guide to Animation: Everything you need to know to get started. Watson-Guptill 3. Richard Williams (2012). The Animator's Survival Kit: A Manual of Methods, Principles, and Formulas for Classical, Computer, Games, Stop Motion, and Internet Animators. Farrar, Straus and Giroux. 4. Tony White(1988).The Animator's Workbook. Watson-Guptill
Course Outcomes:	On completion of the course, students will be able to: 1. Remember terminologies and aspects of computer animation. 2. Apply the different principles of animation to produce still and moving images. 3. Demonstrate and develop 2D animations using different tools. 4. Integrate the concepts of drawing and color theory in animation.

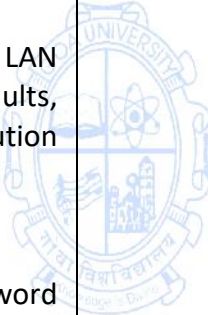
Name of the Programme : Bachelor of Computer Applications
Course Code : CSA-161
Title of the Course : PC Troubleshooting and Networking [Exit Internship Course]
Number of Credits : 4 (2T + 2P)
Effective from AY : 2024-25

Pre-requisites for the Course:	Knowledge of Personal Computer and Programming	
Course Objectives:	1. To understand the PC troubleshooting techniques. 2. To learn the basic concepts of networking. 3. To apply the PC troubleshooting techniques and networking concepts. 4. To analyze the cases of existing network setup and apply it.	
Units	Content	No. of Hours
I	PC Troubleshooting 1. Hardware overview - CPU, RAM, Motherboard, storage devices, etc. 2. Peripherals overview - Monitors, Keyboards, Mouse, Printers, etc. 3. OS overview - OS environments: Windows and Unix / Linux, basic operations and navigation 4. Troubleshooting Fundamentals a. Identifying common PC issues: slow performance, hardware failures, software glitches, etc. b. Introduction to troubleshooting methodologies: isolation, testing, observation 5. Software Troubleshooting a. Diagnostic tools: Task Manager, Event Viewer, Resource Monitor, etc. b. Software installation and removal c. Managing updates and patches d. Web Browser Management e. Firewall & Anti-Virus 6. Hardware Troubleshooting a) Identifying hardware issues: RAM failures, hard drive errors, overheating, printers etc. b) Basic hardware maintenance: cleaning, replacing components c) Introduction to BIOS/UEFI settings	10
II	Networking 1. Introduction to Networking Basics a) Overview of computer networks and their importance b) Introduction to networking terminology and concepts c) Understanding the TCP/IP models 2. Setting Up a Home Network	20

	a) Setting up a basic network environment using consumer-grade routers and switches b) Configuring IP addresses, subnet masks, and default gateways c) Connecting devices to the network (e.g., computers, smartphones, printers) 3. Introduction to Network Protocols a) Hands-on experience with common networking protocols (e.g., TCP, UDP, IP) b) Using packet sniffing tools to analyze network traffic c) Understanding the purpose and structure of Ethernet frames and IP packets 4. Wireless Networking Basics a) Configuring and securing Wi-Fi networks b) Understanding different wireless encryption methods (WEP, WPA, WPA2) c) Troubleshooting common Wi-Fi connectivity issues 5. Network Services Configuration a) Setting up and configuring network services such as DHCP, DNS, and FTP b) Configuring port forwarding and NAT (Network Address Translation) c) Implementing basic firewall rules to control network traffic 6. LAN Design and Troubleshooting a) Designing and implementing a small local area network (LAN) b) Troubleshooting common LAN connectivity issues (e.g., cable faults, IP conflicts) c) Using network diagnostic tools (e.g., ping, traceroute) to identify and resolve network problems 7. Introduction to Network Security a) Basic network security principles and best practices b) Securing network devices with strong passwords and access controls c) Implementing basic security measures such as MAC filtering and disabling SSID broadcast 8. Network Monitoring and Management a) Introduction to network monitoring tools (e.g., Wireshark, Nagios) b) Monitoring network performance metrics (e.g., bandwidth utilization, packet loss) c) Performing basic network troubleshooting and maintenance tasks 9. Introduction to Virtualization and Cloud Computing a) Setting up virtual networks using virtualization	
--	---	--

	platforms (e.g., VMware, VirtualBox) b) Understanding cloud networking concepts and services (e.g., AWS, Azure)	
III	Practical Activities - To be carried out along in sync with the concepts mentioned in Unit I & II respectively. PC Troubleshooting 1) Boot Failure • Identify common causes of boot failure, such as hardware issues, corrupted system files, or misconfigured BIOS settings. • Troubleshoot boot failure by checking hardware connections, performing hardware diagnostics, and accessing BIOS settings to verify boot order and configuration. 2) Blue Screen of Death (BSOD) • Understand common causes of BSOD errors, including driver issues, hardware failures, and software conflicts. • Troubleshoot BSOD errors by analyzing error codes, checking device drivers, and performing memory and disk diagnostics. 3) Slow Performance • Identify factors contributing to slow PC performance, such as insufficient RAM, high CPU usage, or disk fragmentation. • Troubleshoot slow performance by checking resource usage in Task Manager, disabling unnecessary startup programs, and optimizing disk performance with disk cleanup and defragmentation. 4) Internet Connectivity Issues • Troubleshoot network connectivity issues by checking physical connections, verifying network settings, and testing connectivity with other devices. • Use command-line tools like ipconfig and ping to diagnose network problems and resolve issues with DNS resolution or IP address conflicts. 5) Hardware Malfunctions • Identify common hardware malfunctions such as overheating, noisy fans, or malfunctioning peripherals (e.g., keyboard, mouse). • Troubleshoot hardware issues by checking for loose connections, cleaning dust buildup, and replacing	40

	faulty components if necessary. 6) Software Errors • Troubleshoot software errors such as application crashes, freezes, or errors messages. • Use Event Viewer to analyze error logs, update software applications and drivers, and perform malware scans to detect and remove viruses or malware. 7) Peripheral Device Issues • Troubleshoot issues with peripheral devices such as printers, scanners, or external drives. • Check device connections, update drivers, and verify compatibility with the operating system. 8) Data Backup and Recovery • Develop a backup strategy to protect important data from loss due to hardware failure, software errors, or accidental deletion. • Practice data recovery techniques using backup software, file recovery tools, and cloud storage services. 9) System Maintenance • Perform routine system maintenance tasks to optimize PC performance and prevent issues. • Schedule regular updates for the operating system, antivirus software, and device drivers, and perform disk cleanup and defragmentation to maintain disk health. Basic Networking 10) Setting Up a Home Network • Configure a home router: Set up a router with DHCP enabled and configure wireless security. • Connect devices: Connect computers, smartphones, and printers to the network and ensure they can communicate with each other. 11) Introduction to Network Protocols • Packet sniffing with Wireshark: Capture and analyze network traffic to understand protocols like TCP, UDP, and IP. • Ethernet frame analysis: Use Wireshark to examine the structure of Ethernet frames and identify source and destination MAC addresses.	
--	--	--

	12) Wireless Networking Basics • Wi-Fi setup and security: Configure a Wi-Fi network with WPA2 encryption and a strong passphrase. Test connectivity with various devices. • Troubleshoot Wi-Fi issues: Troubleshoot common Wi-Fi problems such as signal interference or connectivity issues. 13) Network Services Configuration • DHCP setup: Configure a DHCP server on a router or server and verify that clients receive IP addresses dynamically. • DNS configuration: Set up a DNS server and configure DNS resolution for local and external domain names. 14) LAN Design and Troubleshooting • LAN setup: Design and implement a small LAN with multiple devices connected through switches. • Troubleshooting scenarios: Simulate LAN connectivity issues such as cable faults, misconfigured IP addresses, or DNS resolution problems. 15) Introduction to Network Security • Password policies: Implement strong password policies on network devices and user accounts. • Firewall setup: Configure basic firewall rules on a router or firewall appliance to control inbound and outbound traffic. 16) Network Monitoring and Management • Bandwidth monitoring: Use network monitoring tools to measure bandwidth utilization and identify bandwidth-intensive applications. • Network troubleshooting: Troubleshoot network issues using diagnostic tools like ping, traceroute, and netstat.	
--	---	--

IV	Case Studies 1. Study the performance of any PC of the College lab, analyze and improve its performance. 2. Analyze any real-world existing networking scenario and case studies, like existing networking of your college labs. Mini - Project <i>Scenario:</i> You have been hired as a network administrator for a small business with approximately 15 employees. The company operates in a single office location and requires a reliable and secure network infrastructure to support its day-to-day operations. Develop a network design and implementation plan for a real-world scenario mentioned above, incorporating all aspects learned throughout the course. Simulate the above plan using a suitable free and open-source simulator like "GNS3" (Graphical Network Simulator-3) OR CISCO Packet Tracer Optional - Prepare for industry-recognized certification (e.g., CompTIA Network+, Cisco CCNA) to enhance employability. Practice exams and hands-on labs to reinforce learning and prepare for certification exams	20
Pedagogy:	Suggested strategies for use to accelerate the attainment of the various course outcomes. 1. A plan is to be developed by the student/s in consultation with the teacher incharge and to be approved. 2. One or methods mentioned below may be used for learning purposes. a. Intensive training / teaching b. Online or offline training (approved by the college or instructor) c. Approved MOOCS Courses d. Workshops - on-campus or off-campus e. Self-learning means & methods f. Enquiry-based learning 3. A work diary to be maintained where all the learning & work carried out to maintained and certified by the teacher incharges. 4. All deliverable & artifacts to be submitted in the college for evaluation and assessments.	

References/ Readings:	Main Reading: 1. Gookin, D. (2021). <i>Troubleshooting and Maintaining Your PC All-in-One For Dummies</i> (4th edition). For Dummies. 2. Kurose, J. F., & Ross, K. W. (2021). <i>Computer Networking: A Top-Down Approach</i> (8th ed.). Pearson Education Ltd. 3. Lowe, D. (2021). <i>Networking All-in-One For Dummies</i>(8th ed.). Wiley. 4. Meyers, M. (2019). <i>CompTIA A+ Certification All-in-One Exam Guide</i> (10th ed.). McGraw-Hill Education. Additional Reading: 1. Beasley, J. S., & Nilkaew, P. (2020). <i>Networking Essentials</i>. Pearson. 2. Donahue, G. A. (2015). <i>Network Warrior</i> (2nd ed.). O'Reilly Media. 3. Mueller, S. (2022). <i>Upgrading and Repairing PCs</i> (10th ed.). Que Publishing 4. Stevens, W. R. (1994). <i>TCP/IP Illustrated, Volume 1: The Protocols</i>(2nd ed.). Addison-Wesley.
Course Outcomes:	On completion of the course, student will be able to 1. Understand the concepts and techniques of PC troubleshooting and basic networking. 2. Apply troubleshooting and networking concepts & strategies and improve the performances. 3. Analyze the performances of PCs and existing networks. 4. Develop a network design for a small group of computers successfully.