



गोंय विद्यापीठ

ताळगांव पठार

गोंय - ४०३ २०६

फोन: +९१-८६६९६०९०४८



(Accredited by NAAC)

Goa University

Taleigao Plateau, Goa - 403 206

Tel : +91-8669609048

Email : registrar@unigoa.ac.in

Website: www.unigoa.ac.in

GU/Acad –PG/BoS -NEP/2023/102/41

Date: 15.06.2023

CIRCULAR

The University has decided to implement the UGC Curriculum and Credit Framework for the Undergraduate Programme (CCFUP) of **Bachelor of Computer Applications/Bachelor of Computer Applications (Honours)** under the National Education Policy (NEP) 2020 from the Academic Year 2023-2024 onwards.

The approved Syllabus of Semesters I and II of the **Bachelor of Computer Applications/Bachelor of Computer Applications (Honours)** Programme is attached.

Principals of Affiliated Colleges offering the **Bachelor of Computer Applications/Bachelor of Computer Applications (Honours)** Programme are requested to take note of the above and bring the contents of this Circular to the notice of all concerned.

(Ashwin Lawande)
Assistant Registrar – Academic-PG

To,

1. The Principals of Affiliated Colleges offering the Bachelor of Computer Applications/Bachelor of Computer Applications (Honours) Programme.

Copy to:

1. The Director, Directorate of Higher Education, Govt of Goa.
2. The Dean, Goa Business School, Goa University.
3. The Vice-Deans, Goa Business School, Goa University.
4. The Chairperson, BOS in Computer Applications.
5. The Controller of Examinations, Goa University.
6. The Assistant Registrar, UG Examinations, Goa University.
7. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

Goa University

Programme Structure for Semester I to VIII Under Graduate Programme

Semester	Major -Core	Minor	MC	AEC	SEC	I	D	VAC	Total Credits	Exit
I	Major- 1 CSA-100 (Problem Solving and Programming) (3T+ 1P)	Minor -1 CSA-111 (Computer System Fundamentals) (4T)	MC-1 CSA-131 (E-Commerce) (3T)		SEC-1 CSA-141 (Office Automation and PC Troubleshooting) (1T + 2P) OR CSA-142 (Python Programming) (1T + 2P)					
II		Minor-2 CSA-112 (Open Source Software) (4T)	MC-2 CSA-132 (Green Computing) (3T)		SEC-2 CSA-143 (Data Analytics using Spreadsheets) (1T + 2P) OR CSA-144 (2D Animation) (1T + 2P)					EXT-1 CSA-161 (Internship) (4)
III	Major- 2 CSA-200 (Programming using C++) (3T+1P) Major- 3 CSA-201 (Database)	Minor -3 CSA-211 (Reasoning Techniques) (3T + 1 Tutorial) OR CSA-212 (Techpreunership)	MC-3 CSA-231 (Cyber Law and Ethics) (3T) OR CSA-232 (Digital Ecosystem)		SEC-3 CSA-241 (Multimedia Applications) (1T + 2P) OR CSA-242 (Search Engine)					

	Management Systems) (3T+1P)	Development) (3T + 1 Tutorial)	(3T) OR CSA-233 (Website Design) (2T+1P) OR CSA-234 (ERP) (2T+1P) OR CSA-235 (Latex) (2T+1P) OR CSA-236 (Multimedia Essentials) (2T+1P)		Optimization) (1T + 2P) OR CSA-243 (3D Animation) (1T + 2P)					
IV	Major-4 CSA-202 (Web App Development) (3P + 1Tutorial) Major-5 CSA-203 (Agile Methodologies) (3T+1P) Major-6 CSA-204 (Object Oriented Concepts) (3T+1P)	Minor-4 VET CSA-221 (Digital Marketing Fundamentals) (3T + 1P) OR CSA-222 (Data Analysis) (3T + 1P)							EXT-2 CSA-261 (Internship) (4)	

	Major-7 CSA-205 (Web Technology) (2T)									
V	Major-8 CSA-300 (UI- UX Design) (3T+1P) Major- 9 CSA-301 (Full Stack Development) (3P + 1Tutorial) Major- 10 CSA-302 (Internet Technologies) (3T + 1P) Major- 11 CSA-303 (Introduction to Functional Programming) (2P)	Minor VET -5 CSA-321 (JavaScript) (3T + 1P) OR CSA-322 (Dashboard Development) (3T + 1P)								

VI	<p>Major-12 CSA-304 (Lambda Calculus) (4T)</p> <p>Major- 13 CSA-305 (Mobile App Development) (3P + 1 Tutorial)</p> <p>Major- 14 CSA-306 (Data Science) (3T + 1P)</p> <p>Major- 15 CSA-307 (Project) (4)</p>	<p>Minor VET-6 CSA-323 (Social Media Marketing & Analytics) (3T + 1P) OR CSA-324 (E-Commerce Applications) (3T + 1P) OR CSA-325 (Modern Frameworks) (3T + 1P)</p>								
VII	<p>Major-16 CSA-400 (Cyber Security) (3T + 1P)</p> <p>Major- 17 CSA-401 (DevOps) (3P + 1 Tutorial)</p> <p>Major- 18 CSA-402 (Software Design Patterns) (3T + 1P)</p> <p>Major- 19 CSA-403</p>	<p>Minor -7 CSA-411 (Project Management) (3T + 1P) OR CSA-412 (Cloud Computing) (3T + 1P) OR CSA-413 (Research Methodology) (3T + 1P)</p>								

	(NLP Applications) (3T + 1P)									
VIII	Major-20 CSA-404 (Educational Technology) (3T + 1P) Major-21 CSA-405 (Information Systems Audit) (3T + 1 Tutorial) Major- 22 CSA-406 (Internet of Things) (3T + 1P) Major- 23 CSA-407 (4T) **	Minor-8 CSA-414 (Interactive Media) (3T + 1P) OR CSA-415 (Game Design) (3T + 1P) OR CSA-416 (Statistical Tools) (3T + 1P) OR CSA-417 (Blockchain Technology) (3T + 1P)								

**** (Title of the Course will be provided separately)**

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: 2023-24

Pre-requisites for the Course:	Nil	
Course Objectives:	<ol style="list-style-type: none">1. To understand the concepts and techniques of problems solving.2. To analyse, 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 75 (45 T + 30 P)
I.	Introduction to Problem Solving <ol style="list-style-type: none">1. Problem Solving Life Cycle – Understanding the Problem Statement, Analyzing the problem, Planning Program design using Hierarchy charts, Expressing Program logic using flowcharts / Pseudocode.2. Structured Programming concept3. Modular Programming - Top-Down design, Bottom-up design, Stepwise Refinement	04
II	Understanding basic problem Solving Tools <ol style="list-style-type: none">1. Algorithms: Definition & its attributes, algorithm constructs, Statements: Input-Output, Decision-Making, & Looping, Examples2. Flowchart: Definition & its attributes, symbols, Statements: Input-Output, Decision-Making & Looping, Module representation, Drawing conventions and standards, Examples.3. Pseudo-code: Definition & its attributes, constructs, and Examples	06
III	Basic Program Structures <p>Data & its types (Integer, Floating-point, Character, String), Constants & Variables, scope, Instructions & its types, how computer stores data, Operators (Arithmetic, Assignment, Relational, Logical, etc), Expressions and Equations, Evaluation of expressions, Keywords.</p> <p>Local and Global Variables, Parameters, Return Values, naming conventions & standards, Understanding literals, syntax and semantics, functions and modules.</p>	06

IV	<p>Basic Sequential Instructions Sequential statements using operators, constants, variables, operands, expressions and equations. Activity: Design algorithms of at-least 3 basic problems to apply the concept learned in the unit. Represent it using flowchart and pseudo-code.</p>	06
V	<p>Problem Solving with Decisions The Decision Logic Structure, Multiple If/Then/Else Instructions, Using Straight-Through Logic, Using Positive & Negative Logic, Logic Conversion, Decision Tables, Case Logic Structure Activity: Design algorithms of at-least 3 basic problems to apply the concept learned in the unit. Represent it using flowchart.</p>	06
VI	<p>Problem Solving with Loops The Loop Logic Structure, Incrementing, Accumulating, While/While End, Repeat/Until, Automatic-Counter Loop, Nested Loops, Indicators (flags). Activity: Design algorithms of at-least 3 basic problems to apply the concept learned in the unit. Represent it using flowchart.</p>	06
VII	<p>Understanding functions Functions: Definition and its need & constructs, designing simpler functions, function communication using arguments & 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 call. Base case and recursive case. Activity: Design algorithms of at-least 3 basic problems to apply the concept learned in the unit. Represent it using flowchart.</p>	06
VIII	<p>Problem Solving with Arrays Arrays Concepts: One dimensional Arrays, Creating, iterating, accessing and modifying array elements. Concept of Strings, String as array of characters. Activity: Design algorithms of at-least 3 basic problems to apply the concept learned in the unit. Represent it using flowchart.</p>	03
IX	<p>Debugging & Documentation Definition, Types, Need and how to do it.</p>	02
Practical Work	Using any suitable programming language (eg C, C++, etc.), the concepts learnt in the units from I to IX are required to	Practical Hours (30)

	implemented practically. The broad area of practical problems are mentioned / suggested below.	
<p>Week 1 & 2 [These practicals should be done using pen & paper and using buddy learning strategy]</p>	<ol style="list-style-type: none"> 1. For each of the following tasks, write a set of numbered, step-by-step instructions (a solution) so complete that another person could perform the task without asking questions. Define the knowledge base of this person by listing what you expect the person to know in order 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. <ol style="list-style-type: none"> a. Make a cup of cocoa. b. Sharpen a pencil. c. Walk from the classroom to the student lounge, your dorm, or the cafeteria. d. Start a car (include directions regarding what to do if the car doesn’t start). e. Get a glass of water from your kitchen. f. 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 Hours
Week 3 & 4	<ol style="list-style-type: none"> 3. Basic Program Structures <ul style="list-style-type: none"> ● At-least 10 basic programming problems related to Unit III 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 Hours
Week 5 & 6	<ol style="list-style-type: none"> 4. Basic Sequential Instructions <ul style="list-style-type: none"> ● At-least 08 programming problems related to Unit IV 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 Hours
Week 7 & 8	<ol style="list-style-type: none"> 5. Problem Solving with Decisions <ul style="list-style-type: none"> ● At-least 08 programming problems related to Unit V 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 Hours

<p>Week 9 & 10</p>	<p>6. Problem Solving with Loops</p> <ul style="list-style-type: none"> ● At-least 06 programming problems related to Unit VI 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. 	<p>04 Hours</p>
<p>Week 11 & 12</p>	<p>7. Understanding functions</p> <ul style="list-style-type: none"> ● At-least 08 programming problems related to Unit VII 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. 	<p>04 Hours</p>
<p>Week 13 & 14</p>	<p>8. Problem Solving with Arrays</p> <ul style="list-style-type: none"> ● At-least 06 programming problems related to Unit VIII 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. 	<p>04 Hours</p>
<p>Week 15</p>	<p>9. Debugging & Documentation</p> <ul style="list-style-type: none"> ● Debug & Document at-least 06 problems which you have programmed from week 07 onwards. ● More programs may be given to the learners to complete and practice as part of their Practice Work. 	<p>02 Hours</p>
<p>Pedagogy:</p>	<p>Suggested strategies for use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. 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 <ol style="list-style-type: none"> 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, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse 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 atleast one assignment (equivalent to 50% assignment weightage) where they can complete atleast one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 	

References/ Readings:	<ol style="list-style-type: none"> 1. Maureen Sprankle and Jim Hubbard, Problem Solving and Programming Concepts, Pearson Education India 9th edition (2013) 2. S.Kuppuswamy, S.Malliga, C.S.Kanimozhi Selvi, K.Kousalya. Problem Solving and Programming. 2019, Tata McGraw Hill. 3. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg ISBN:9788131500941, Cengage Learning India 4. Introduction to algorithms – Cormen, Leiserson, Rivest, Stein Ritchie,ISBN:9788120305960, PHI Learning 5. How to Solve it by Computer, R.G. Dromey, ISBN: 9788131705629, Pearson Education <p>Article in Online Encyclopedia</p> <ol style="list-style-type: none"> 1. https://code.world/ [Accessed: April 15, 2023]. 2. https://raptor.martincarlisle.com/ [Accessed: April 15, 2023].
Course Outcomes:	<p>On completion of the course, students will be able to –</p> <ol style="list-style-type: none"> 1. Understand the ways and stages of Problem Solving. 2. Understand basic computing concepts, algorithm design, flowchart design, programming constructs and debugging. 3. Apply the problem solving & programming concepts in designing solution to simpler problems. 4. Code and implement a well-structured programming logic using a suitable programming language.

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: 2023-24

Pre-requisites for the Course:	Nil	
Course Objectives:	<ol style="list-style-type: none">1. To understand the basic concepts of Operating System and Computer Architecture2. To understand the concepts of organization and functioning of Computer System.3. To design Algorithms and flowcharts for better understanding and documentation for accurate implementation of the problem.4. To learn and use efficient programming constructs.	
Content		No of hours (60)
Unit	Topic	
I	<p>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, Generations of Computer, Von Neumann Architecture.</p> <p>Computer Components and Functions – Instruction Cycle with and without interrupts, Multiple interrupts, Bus Interconnections. Number System –Conversion (Binary, Decimal, Octal, Hexa-Decimal)</p>	10
II	<p>Central Processing Unit Computer Arithmetic ALU, Integer representation, Integer Representation, Addition, subtraction. Instruction sets, characteristics & Functions, Addressing modes and formats. CPU structure and function. Processor Generation-8086,Pentium I-IV,i1-i7</p>	05
III	<p>Processes & Process Management Process- Definition, Process Control Block, Process States, Operations on Process. Threads and Microkernels – Definition, Multithreading, Model Process Scheduling - Introduction to the Concept, Scheduling Criteria Scheduling Algorithms.</p>	15

	<p>Concurrency/ Process Coordination – Synchronization Principles, Mutual Exclusion, The Critical-Section Problem, Peterson’s Solution</p> <p>Deadlock – Principles, Deadlock Handling Methods, Prevention, Avoidance, Detection, Recovery from Deadlock</p>	
IV	<p>Memory Management Memory Management Concepts – Swapping, Contiguous Memory Allocation, Paging, Page Table, Segmentation.</p> <p>Virtual Memory – Introduction, Demand Paging, Page Replacement Frames Allocation, Thrashing Cache Memory, Internal Memory (Semiconductor, Main Memory, RAM), External Memory (Magnetic Disk, RAID, Optical memory)</p>	10
V	<p>Input/ Output and File System File System Concepts, File Organization and Access Methods, Directory Structure, File Sharing</p> <p>I/O Management - I/O devices, I/O Hardware, Organization of I/O, I/O Buffering, Disk Scheduling- Algorithms, I/O Modules, I/O techniques (programmed, interrupt driven and DMA), Operating System Support.</p>	14
VI	<p>Control Unit Structure of the Control Unit, Functioning of the Control Unit, Micro programmed control</p>	6
Pedagogy:	<p>Suggested strategies to use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. 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 <ol style="list-style-type: none"> 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, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse 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. 	

	<ol style="list-style-type: none"> 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 at least one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.
<p>References/ Readings:</p>	<ol style="list-style-type: none"> 1. Pradipt Sinha and Priti Sinha, "Computer Fundamentals", 6th Edition, BPB Publications, 2016 2. Abraham Silberschatz and Peter Baer Galvin, "Operating System Concepts", 7th Edition, Pearson Education, 2007. 3. William Stallings, "Operating Systems", 6th Edition, Pearson Education, 2010 4. Computer Organization and Architecture (7th Edition): William Stallings, Prentice-Hall. 5. Computer System Architecture: Morris Mano, Prentice-Hall. <p>E-Books:</p> <ol style="list-style-type: none"> 1. Operating Systems Guide: by Tim Bower 2. Operating Systems Course Notes: by Dr. John T. Bell 3. Schaum's Outline of Operating Systems (Schaum's Outline Series) [Kindle Edition] by J. Archer Harris. 4. Computer Organization: TMH, Ace series. 5. Computer Organization and Architecture by William Stallings, 8th Edition, Prentice-Hall.
<p>Course Outcomes:</p>	<p>On completion of the course, students will</p> <ol style="list-style-type: none"> 1. Understand the basic structure and functioning of the Computer 2. Get the understanding of various functions and characteristics of operating systems. 3. Understand basic concepts of Computer Organisation & Architecture. 4. Understand basic understanding of resource allocation, functioning of the I/O modules, primary and secondary memory, and Control Unit.

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: 2023-24

Prerequisite for the Course :		Nil	
Course Objectives :		1.Explain the benefits of Open Source software 2.Understand the social impact of Open Source software 3.Understanding tools and techniques for creating and managing open source communities of practice 4.Understanding Open Source software ecosystem	
Unit	Title	Content	No of Hours (60)
I	Introduction to Open source Software	Introduction <ul style="list-style-type: none">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 <ul style="list-style-type: none">Open Source History, Initiatives, Principle 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 Social Impact <ul style="list-style-type: none">Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government.	20
II	Contribution to Open Source Projects	Case Studies <ul style="list-style-type: none">Example Projects: Apache web server, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, wordpress, GCC, GDB, github, Open Office. Study: Understanding the developmental models, licensings, mode of funding, commercial/non-commercial use. Open Source Hardware, Open Source Design, Open source Teaching. Open source media. Collaboration, Community and Communication Contributing to Open Source Projects	20

		<ul style="list-style-type: none"> ● Introduction to github, interacting with the community on github, Communication and etiquette, testing open source code, reporting issues, contributing code. ● Introduction to wikipedia, contributing to Wikipedia Or contributing to any prominent open source project of student's choice. ● Starting and Maintaining own Open Source Project 	
III	Open Source Ecosystem	<p>Understanding Open Source Ecosystem</p> <ul style="list-style-type: none"> ● 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 	20
Pedagogy:		<p>Suggested strategies to use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. 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 <ol style="list-style-type: none"> 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, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse 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 (equivalent to 50% assignment weightage) where they can complete at least one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 	
References/ Readings:		<p>Text book:</p> <ol style="list-style-type: none"> 1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006 2. The official Ubuntu Book, 8th Edition <p>Article in Online Encyclopedia [All accessed on: April 15, 2023]</p> <ol style="list-style-type: none"> 3. The Linux Documentation Project: http://www.tldp.org/ 4. Docker Project Home: http://www.docker.com 	

	<ol style="list-style-type: none">5. Linux kernel Home: http://kernel.org6. Open Source Initiative: https://opensource.org/7. Linux Documentation Project: http://www.tldp.org/8. Wikipedia: https://en.wikipedia.org/9. https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia10. Github: https://help.github.com/11. The Linux Foundation: http://www.linuxfoundation.org/
Course Outcomes:	<p>At the end of the course, the student will be able to -</p> <ol style="list-style-type: none">1. Understand the significance of Open Source software practices and guidelines2. Understand knowledge of Open Source ecosystem, its use, impact and importance.3. Use Open Source methodologies, case studies in real life applications.4. Collaborate and contribute to Open Source Projects

Name of the Programme: Bachelor of Computer Applications

Course Code: CSC-131

Title of the Course: E-Commerce

Number of Credits: 3T

Effective from AY: 2023-24

Pre-requisite for the Course:	Nil	
Course Objectives:	1. To give fundamental understanding of e-commerce 2. To instill idea of convergence of business relationship through technologies. 3. To understand the application of e-commerce in B2B and B2C modal. 4. To identify, define and differentiate the various modes and risks of electronic commerce.	
Units	Content	No of hours 45
1	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	5 hours
2	E-Commerce Web-sites Websites as marketplace, Role of website in B2C e-commerce, Website design principles, Alternative methods of customer communication such as e-mail, Email etiquette and e-mail security	5 hours
3	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, Search Engine Optimization(SEO), Search Engine Marketing(SEM), Social Media Marketing(SMM), Web Analytics	10 hours
4	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	5 hours
5	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	4 hours
6	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,	7 hours

	Characteristic features of EDI service arrangement, EDI architecture and standards	
7	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, Risk management options for e-payment systems	5 hours
8	Security Issues in E-Commerce Risks of e-commerce,Types and sources of threats; Security tools, Risk management approaches	4 hours
Pedagogy:	<p>Suggested strategies for use to accelerate the attainment of the various course outcomes.</p> <p>10. 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</p> <ol style="list-style-type: none"> Video/Animation to explain various concepts. Collaborative, Peer, Flipped Learning etc. <p>11. Ask at least three HOT (Higher-Order Thinking) questions in the class, which promotes critical thinking.</p> <p>12. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse information rather than simply recall it.</p> <p>13. Introduce Topics in manifold representations.</p> <p>14. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.</p> <p>15. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding</p> <p>16. To promote self-learning give atleast one assignment (equivalent to 50% assignment weightage) where they can complete atleast one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations.</p>	
References/ Readings:	<p>Reference Books:</p> <ol style="list-style-type: none"> P. T. Joseph, E-Commerce: An Indian Perspective Paperback, PHI Learning, 5th Edition, 2015 V. Rajaraman, Essentials of E-Commerce Technology, PHI Learning, 2015 Revised Edition Kalakota, Ravi and Andrew.Whinston, Frontiers of Electronic Commerce, Pearson Education, 2015 Revised Edition Kamlesh N. Agarwala, Amit Lal and Deeksha Agarwala, Business on the Net: An Introduction to the Whats and Hows of E Commerce, Macmillan India Ltd, 2000 Diwan, Pragand Sunil Sharma, Electronic Commerce- A Manager's Guide to E Business,Vanity Books International, Delhi. 	

	<p>6. Fitzgerald, Business Data Communication Network, McGrawHill, 1998.</p> <p>7. Dishek J. Mankad, Understanding Digital Marketing: Strategies for online success, 2019</p> <p>8. NPTELResources: https://nptel.ac.in/content/storage2/courses/106108103/pdf/PPTs/mod13.pdf</p>
<p>Course Outcomes:</p>	<p>On completion of the course students will be able to</p> <ol style="list-style-type: none"> 1. Understand the foundation of e-commerce and e-commerce websites. 2. Explain the basics of online marketing and e-commerce applications. 3. Compare B2B and B2C e-commerce models. 4. Explain electronic payment system, the security issues, and security mechanism in e-commerce applications.

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: 2023-24

Prerequisite for the Course :		Nil	
Course Objectives :		<ol style="list-style-type: none">1. To understand Green IT concepts and meeting standards set for Green Computing2. To comprehend Green IT from the perspective of hardware, software, storage, and networking at the enterprise level.3. To understand Green Initiatives and future of Green IT	
Unit	Title	Content	No of Hours (45)
I	Trends and Reasons to Go Green	<ul style="list-style-type: none">● Overview and Issues● Current Initiatives and Standards● Consumption Issues - Minimizing Power Usage, Cooling	05
II	Introduction to Green IT	Green IT <ul style="list-style-type: none">● Holistic Approach to Greening IT● Awareness to Implementation<ul style="list-style-type: none">▪ Green IT Trends▪ Green Engineering Greening by IT <ul style="list-style-type: none">● Using RFID for Environmental● Sustainability● Smart Grids● Smart Buildings and Homes● Green Supply Chain and Logistics● Enterprise-Wide Environmental Sustainability	08
III	Green Hardware and Software	Green Hardware <ul style="list-style-type: none">● Introduction● Life Cycle of a Device or Hardware● Reuse, Recycle, and Dispose Green Software <ul style="list-style-type: none">● Introduction● Energy-Saving Software Techniques Changing the way we work <ul style="list-style-type: none">● Going Paperless	08

<p>IV</p>	<p>Green Data Centres and Storage</p>	<p>Green Data Centres</p> <ul style="list-style-type: none"> ● Data Centre IT Infrastructure ● Data Centre Facility Infrastructure: Implications for Energy Efficiency ● IT Infrastructure Management ● Green Data Centre Metrics <p>Green Data Storage</p> <ul style="list-style-type: none"> ● Introduction ● Storage Media Power Characteristics ● Energy Management Techniques for Hard Disks ● System-Level Energy Management <p>Green Networks and Communications</p> <ul style="list-style-type: none"> ● Introduction ● Objectives of Green Network Protocols ● Green Network Protocols and Standards 	<p>08</p>
<p>V</p>	<p>Enterprise Green IT Strategy</p>	<ul style="list-style-type: none"> ● 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 	<p>6</p>

VI	Managing and Regulating Green IT	<p>Managing Green IT</p> <ul style="list-style-type: none"> ● Introduction ● Strategizing Green Initiatives ● Implementation of Green IT ● Information Assurance ● Communication and Social Media <p>Regulating Green IT</p> <ul style="list-style-type: none"> ● 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 <p>The Future of Green IT</p> <ul style="list-style-type: none"> ● Green Computing and the Future ● Megatrends for Green Computing ● Tele-presence Instead of Travel ● Tele-commuting Instead of Commuting ● Deep Green Approach 	10
Pedagogy:		<p>Suggested strategies to use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. 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 <ol style="list-style-type: none"> 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, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse 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 atleast one assignment (equivalent to 50% assignment weightage) where they can complete atleast one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 	

References:	<ol style="list-style-type: none"> 1. Toby Velte, Anthony Velte, Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line, McGraw Hill Education, 2008 2. San Murugesan, G. R. Gangadharan, Harnessing Green IT: Principles and Practices, Wiley, 2013 3. Bud E. Smith, Green Computing-Tools and Techniques for saving energy, money and resources, Auerbach Publications 4. Mark G. O'Neill, Green IT for Sustainable Business Practice, BCS, The Chartered Institute for IT 5. Jason Harris, Green Computing and Green IT Best Practices, Emereo Pty Ltd
Course Outcome:	<p>At the end of the course, the students will be able to</p> <ol style="list-style-type: none"> 1. Understand Green Computing & Green IT fundamentals. 2. Acquire knowledge on Green Hardware and Software, Green Data Centres and Green IT Strategies 3. Understand the concept of green compliance 4. Express the understanding of green initiatives.

Name of the Programme: Bachelor of Computer Applications

Course Code: CSA-141

Title of the Course: Office Automation and PC troubleshooting

Number of Credits: 1T + 2P

Effective from AY: 2023-24

Prerequisite for the Course :	Nil		
Course Objectives :	<ol style="list-style-type: none">1. Understand the basics of office automation software and its applications.2. Develop proficiency in using word processing, spreadsheet, and presentation software.3. Identify common hardware and software issues in a PC.4. Diagnose and troubleshoot common PC issues.5. Develop skills in maintaining and optimizing the performance of a PC.		
#	Title	Content	No of Hours
	THEORY		15
1.	Introduction to Office Automation	Understanding office automation software and its applications Types of office automation software Microsoft Office Suite Google Workspace	1
2.	Word Processing	Introduction to Microsoft Word Creating and formatting documents Working with templates Mail merge and labels Collaboration tools	3
3.	Spreadsheets	Introduction to Microsoft Excel Creating and formatting spreadsheets Working with formulas and functions Charts and graphs Collaboration tools	3
4.	Presentation Software	Introduction to Microsoft PowerPoint Creating and formatting presentations Working with images, videos, and animations Collaboration tools	2
5.	Email Management	Introduction to Email Setting up and configuring email accounts Composing and sending emails Managing Email Accounts	1

6.	Internet and Web Browsers	Introduction to the Internet Web Browsers Searching the Internet Configuring web browser settings	1
7.	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	4
PRACTICAL			60
<ol style="list-style-type: none"> 1. Practical on Google Workspace 2. Practical on Word Processing <ul style="list-style-type: none"> - Bulleted and numbered list, headers and footers, page numbering. - Creation of tables - Mail Merge 3. Practical on Spreadsheet <ul style="list-style-type: none"> - Formatting of cells, rows and columns - Presenting data with charts - Working with formulae 4. Practical on Presentation software <ul style="list-style-type: none"> - Usage of text, images and animation for presentation - Adding slide transition, custom animation, set up show. - Creating graphs in presentation. 5. Practical on Internet browsing, downloading files, knowing secure browsing. 6. Practical on Email account creation, sending emails, attachments. 7. Practical on PC troubleshooting <ul style="list-style-type: none"> - Understanding PC components and PC assembling, formatting, fragmentation and installation of Operating system and configuration of different types of software. - Installation of different hardware devices, configuring Printers - Identifying issues with hardware devices and troubleshooting. - Network setup of two or more PCs. - System protection, antivirus and firewall 			

<p>Pedagogy:</p>	<p>Suggested strategies to use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. 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 <ol style="list-style-type: none"> 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 analyse 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 (equivalent to 50% assignment weightage) where they can complete atleast one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 8. Activity/ Practical Based Learning (Suggested Activities in Class) <ol style="list-style-type: none"> a. Real world problem solving using group discussion. E.g., designing poster for road safety etc., b. Demonstration of solution to a problem through design. 9. Demonstration of simple project and motivating the students to develop similar type of projects.
<p>References/Readings:</p>	<ol style="list-style-type: none"> 1. Discovering Computers 2022: Digital Technology, Data, and Devices by Misty E. Vermaat 2. A+ Guide to IT Technical Support (MindTap Course List) by Jean Andrews 3. Shelly, G. B., &Vermaat, M. E. (2017). Microsoft Office 365 & Office 2016: Introductory. Cengage Learning. 4. Russel, C., & Hoque, M. R. (2018). Google Workspace for Dummies. John Wiley & Sons. 5. Meyers, M. (2017). CompTIA A+ Certification All-in-One Exam Guide, Ninth Edition (Exams 220-901 & 220-902). McGraw Hill Professional.
<p>Course Outcomes:</p>	<p>On completion of the course, students will be able to</p> <ol style="list-style-type: none"> 1. To create and format documents, create and format tables and mail merge. 2. Understand the use and various functions of spreadsheets 3. To apply the knowledge of tools to create effective presentations. 4. Understand PC assembling and troubleshooting

Name of the Programme: Bachelor of Computer Applications

Course Code: CSA-142

Title of the Course: Python Programming

Number of Credits: 1T +2P

Effective from AY: 2023-24

Prerequisite for the Course :		Nil	
Course Objectives :		<ol style="list-style-type: none">1. To understand simple Python programs.2. To develop Python programs with conditionals and loops.3. To define Python functions.4. To use Python data structures – lists, tuples, dictionaries.5. To deal with input/ output files in Python.6. To understand application areas of Python.	
#	Title	Content	No of Hours (75)
		THEORY	15
I	Introduction to Python	<ul style="list-style-type: none">• Data values and types: int, float, Boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments.	4
II	Program Flow Control	<ul style="list-style-type: none">• Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, immutability, string functions and methods, string module; Lists as arrays.	4
III	List, Tuple and Dictionary	<ul style="list-style-type: none">• Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension.	3
IV	Files, Modules, Packages	Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages.	3

V	Application Areas	<ul style="list-style-type: none"> Google Translate, Sentiment Analysis: Analyse Facebook data, Image processing, Page rank. 	1
PRACTICAL		60	
	<ol style="list-style-type: none"> 1. Installation & IDE 2. Program to understand variables & different data types 3. Program to perform basic Input and Output operations 4. Program to demonstrate operations (Arithmetic, assignment, comparison) 5. Math, Strings, and Variables 6. Program to demonstrate Conditional Statements 7. Program to demonstrate setting precedence 8. Program to perform casting data types. 9. Program to demonstrate Control Structures 10. Repetition Structures- Program to demonstrate while loop and for loop 11. Program to demonstrate Break and Continue statements 12. Program to create custom Functions 13. Program to demonstrate local and global variables 14. Program to demonstrate arguments and return values 15. Program to perform list manipulation 16. Program to demonstrate Sets and its methods 17. Write a Tuple and perform sequence unpacking 18. Program to demonstrate key value pairs in dictionaries 19. Program to demonstrate recursive function 20. Program to perform File Input and Output 21. Program to demonstrate exception handling 		
Pedagogy:	<p>Suggested strategies to use to accelerate the attainment of the various course outcomes:</p> <ol style="list-style-type: none"> 1. 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 2. Video/Animation to explain various concepts. 3. Collaborative, Peer, Flipped Learning etc. 4. Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse information rather than simply recall it. 6. Introduce Topics in manifold representations. 		

	<ol style="list-style-type: none"> 7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 9. To promote self-learning give atleast one assignment (equivalent to 50% assignment weightage) where they can complete atleast one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 10. One internal practical exam will be conducted as a part of internal evaluation. 11. Practical shall be performed in the laboratory as indicated in the syllabus. 12. A softcopy of e-journal shall be maintained clearly mentioning the name of the experiment and other required information.
References:	<ol style="list-style-type: none"> 1. John V Guttag, Introduction to Computation and Programming Using Python“, Revised and expanded Edition, MIT Press , 2013 2. Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016, 1st Edition 3. Timothy A. Budd, Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2015. 4. Kenneth A. Lambert, Fundamentals of Python: First Programs, CENGAGE Learning, 2012, New Edition 5. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, Updated for Python 3, Shroff/O’Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/) 6. Guido van Rossum and Fred L. Drake Jr, An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
Course Outcomes:	<p>At the end of the course, the students will be able to -</p> <ol style="list-style-type: none"> 1. Describe the datatypes, various Control Structures used in Python. 2. Decompose a Python program into functions and recursive functions. 3. Represent compound data using Python lists, tuples, and dictionaries. 4. Understanding use of files and packages in Python Programs.

Name of the Programme: Bachelor of Computer Applications

Course Code: CSA-143

Title of the Course: Data Analytics using Spreadsheets

Number of Credits: 1T + 2P

Effective from AY: 2023-24

Pre-requisites for the Course:	Nil		
Course Objectives:	<ul style="list-style-type: none">● To be familiar and understand spreadsheet software.● To be familiar and learn basic and advanced functions in any spreadsheet.● To learn data analysis and data visualization with charts and pivot tables.● To be familiar with power query in spreadsheets and learning joins.● To understand data analysis tools and the functions used.		
Sr. No.	Title	Content	No. of Hours
THEORY			15 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	1
II	Formulas and Functions	Sum, Average, Min, Max, count, IF, nested IF, using IF with AND OR formulas, COUNTIF, SUMIF, AVERAGEIF formulas, TEXT functions	2
III	Advanced Functions	Vlookup function, match function, index function, date and time functions, maths functions, financial functions	2
IV	Data Analysis	Conditional formatting, What if analysis using data table, Goal seek, scenario manager, Linear regression	2
V	Charts and Visualization	Data storytelling tips, Introduction to charts, types of charts, uses and benefits, Understanding Pivot tables, Pivot table tips and tricks	2
VI	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 outer join and inner join in power query, left outer join and right outer join in power query, Left anti join and right anti join in power query	3

VII	Dashboard reporting in spreadsheets	Understanding how to create dashboard in spreadsheets, Sales Analytical Dashboard using Data Analysis Expressions (DAX) & Visualization, creating a simplified GANTT chart with AND function	2
VIII	Data Analysis tools	ANOVA, Correlation, Covariance, regression, sampling, t-test, z-test and histograms	1
PRACTICAL			60 hours
1	Practical on introduction to spreadsheet using simple tabular data and formatting using paste special, absolute and relative cell references, calculating sum, average, min, max, count and percentage.		4
2	Practical using IF, NESTED IF, SUMFIF, AVERAGEIF, COUNTIF		4
3	Practical on advanced functions		8
4	Practical on conditional formatting, what if analysis using Goal seek, scenario manager and linear regression		4
5	Practical on different types of charts and pivot table with suitable examples		8
6	Practical on Power query, DAX and different types of joins with suitable data.		12
7	Creating dashboard and gantt chart in spreadsheet using suitable examples		8
8	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	<p>Suggested strategies to use to accelerate the attainment of thevarious course outcomes.</p> <ol style="list-style-type: none"> 1. Lecture method need not be only a traditional lecture method, but alternativeeffective teaching methods could be adopted to attain the outcomes. You may use <ol style="list-style-type: none"> 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, whichpromotes critical thinking. 3. Adopt ProblemBased Learning (PBL), which fosters students' Analytical skills,develop design thinking skills such as the ability to design, evaluate, generalize, 		

	<p>and analyse information rather than simply recall it.</p> <ol style="list-style-type: none"> 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 (equivalent to 50% assignment weightage) where they can complete at least one MOOCs (certificate or equivalent) course out of lecture hour. Test their understanding through quizzes or presentations. 8. One assignment in the form of mini-project collecting data and using analytic tools may be given to the students. 	
References/Readings	<ol style="list-style-type: none"> i) Kenneth N Berk, Data Analysis with Microsoft Excel ii) Microsoft Excel 2019 Data Analysis And Business Modeling, Sixth Edition, Microsoft. 	
Course Outcomes	<p>On completion of the course learners will be able to:-</p> <ol style="list-style-type: none"> 1. Understand the basics of spreadsheets and advanced functions 2. Apply data analysis and data visualization using charts and pivot tables. 3. Apply the knowledge of power query and DAX in spreadsheets. 4. Apply data analysis tools and solve simple real life data analysis applications. 	

Name of the Programme: Bachelor of Computer Applications

Course Code: CSA-144

Title of the Course: 2D Animation

Number of Credits: 03 (1T + 2P)

Effective from AY: 2023-24

Prerequisite for the Course :	<ul style="list-style-type: none">Basic concepts of graphics design.
Course Objectives :	<ol style="list-style-type: none">Familiarize with various approaches, methods and techniques of Animation Technology.Study the basics of color theory and graphics.Master traditional & digital tools to produce stills and moving images.Develop expertise in life-drawing and related techniques.Apply laws of human motion and psychology in 2-D characters.Apply Audio and Video Production Techniques to an Animation Project.

#	Title	Content	No of Hours (75)
		THEORY	15
I	Introduction to Animation	<ul style="list-style-type: none">Introduction to Animation, Terms used in AnimationTypes of Animation- Cel (Celluloid) Animation, 2D Animation, 3D Animation,Motion Graphics, Stop Motion.Animation Techniques- Hand-drawn animation, Cut-out animation, Model animation or Stop motion animation, Computer animation or computer generated imagery.Equipment required for animation- Pen tablet, Graphic tablet, Artist glove, Ergo stand, Flex arm.	3
II	Principles of Animation	<ul style="list-style-type: none">Disney's twelve basic principles of animation- Squash Principles of Animation 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 string slices, immutability, string functions and methods, string module; Lists as arrays.	3
III	Fundamentals of Drawing and Design	<ul style="list-style-type: none">Basic Drawing techniques, Concepts of Visualization- Perspective drawing, Illustration and Sketching techniques, Basic Shapes and Sketching Techniques, Modelling digital objects that one can find reference for in	2

		the real world, Modelling hard surface, Developing Animation Character, shading objects and techniques.	
IV	Color Theory and Graphics	<ul style="list-style-type: none"> ● Color fundamentals- primary colors, secondary colors, Tertiary Colors, Color balance, Properties of color-Hue, Reflective Value, Tints and Shades, Saturation, ● Color tone – Intensity ● Color swatches, Color Charts, Safety Colors & Industrial Identification - Additive ● Color System (RGB) - Subtractive Color System (CMYK). ● Vector and Raster graphics - Overlapping shapes, Reshaping lines and shape ● outlines - Snapping (object snapping, pixel snapping, snap alignment), ● Working with color, strokes and fills. 	3
V	2D Animation	<p>2D Animation tools processing</p> <p>2. 2D animation software paradigms-Scripting & Storyboarding, Usage of tools for Digital Painting and vector drawings, How to develop a character and background creation, Usage of timeline and its purpose, Creation of symbols, Onion skinning.</p> <p>Basics of 2D Animation</p> <p>3. Introduction to 2D Animation, 2D motion graphics, Incorporating images into 2D animation, Incorporating sound into 2D animation Exporting your work to various formats-Still image, GIF, Video, Flash.</p>	3
VI	Motion Data Processing	4. History of motion capture, recording actions of human actors, and using that information to animate digital character models in 2D computer animation	1
	PRACTICAL		60
	<p>List of suggested Practical:</p> <ol style="list-style-type: none"> 1. Flip Book Drawing simple flip book with minimum 10 pages 2. Frame by frame animation Creating simple frame by frame animation for a short animation (maximum 20 sec with color drawings and background. 3. Tween Creating simple animation with shape, classic & motion tweening. 4. Ball animation 		

	<p>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.</p> <p>5. Character drawing Drawing simple character with pen tool or shape tool, Preparing the character for animation, dividing each body parts into symbol and creating motion</p> <p>6. Human/ Animal walk cycle Drawing cycle sheet for an animal walk cycle, Creating four different types of walk cycle (jump, run, tip toe, crawl)</p> <p>7. Mini project Creating a short animation film</p>	
--	--	--

<p>Pedagogy:</p>	<p>Suggested strategies to use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. 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 <ol style="list-style-type: none"> a. Video/Animation to explain various concepts. b. Collaborative, Peer, Flipped Learning etc. 2. Introduce Topics in manifold representations. 3. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them. 4. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding 5. To promote self-learning give at least one assignment (equivalent to 50% assignment weightage) where they can complete at least one MOOCs (certificate or equivalent) course out of lecture hour. 6. Course delivery pattern, evaluation scheme, prerequisite shall be discussed at the beginning. 7. Lectures preferably to be conducted with the aid of multi-media projector, black board, group activities, cases, etc. 8. One internal written/practical exam would be conducted as a part of internal theory evaluation. 9. One assignment based on the course content may be given to the students to evaluate how learning of objectives was achieved. 10. The course has a separate laboratory, where students have an opportunity to build an appreciation for the concepts being taught in this course. 11. Mini-Project may be given as part of assessment
------------------	--

	12. Suggestive software's for 2d animation: pencil 2d, adobe flash/animate, synfig
References:	<ol style="list-style-type: none"> 1. Mary Murphy, Beginner's Guide to Animation: Everything you need to know to get started, Watson-Guption 2. Chris Patmore, The Complete Animation course, Barons Educational Series (New York) 3. Stephen cavalier, The world history of animation, Disney animation, Disney editions 1, 9 Sep 2011. 4. Richard Williams, The Animator's Survival Kit : A Manual of Methods, Principles and Formulas for Classical, Computer, Games, Stop Motion and Internet Animators. Expanded Edition 5. Alberto Menache, Understanding Motion Capture for Computer Animation, The Morgan Kaufmann Series in Computer Graphics Second Edition
NPTEL	6. Introduction to Computer Graphics : https://nptel.ac.in/courses/106/102/106102065/
Course Outcome:	<p>On completion of the course learners will be able to:-</p> <ol style="list-style-type: none"> 1. Define terminologies and aspects of computer animations. 2. Use different tools and techniques of animating graphics. 3. Implement the concepts of colors, shapes and digital imagery. 4. Design and develop 2D and 3D animations using different tools.