Dr. BABASAHEB AMBEDKAR MARATHWADA UNIVERSITY,
CHHATRAPATI SAMBHAJINAGAR.

CIRCULAR NO.SU/ Sci./College/NEP-2020/105/2024

It is hereby inform to all concerned that, In continuation circular No.SU./Revised B.Sc./NEP/72/2024/25588-96 dated 29.04.2024, the revised syllabi prepared by the Board of Studies/Ad-hoc Boards and recommended by the Dean, Faculty of Science & Technolgy, the Academic Council at its meeting held on 08 April 2024 has accepted **the following Revised B.Sc. Course Structure & Curriculum** as per direction by the State Government dated on 13 March 2024 under the Faculty of Science & Technology (as per National Education Policy – 2020) run at the Affiliated Colleges, Dr.Babasaheb Ambedkar Marathwada University as appended herewith.

Sr.No.	Courses	Semester
1	Non Conventional & Conventional Enery (Single Major)	Ist and IInd semester
2	Home Science (Single Major)	Ist and IInd semester
3	Bachelor of Computer Application (Single Major)	Ist and IInd semester
4	Computer Science (Single Major)	Ist and IInd semester
5	Data Science (Single Major)	Ist and IInd semester
6	Inforamtion Technology (Single Major)	Ist and IInd semester
7	Networking and Multimedia (Single Major)	Ist and IInd semester
8	Automobile Technology (Single Major)	Ist and IInd semester
9	Forensic Science (Single Major)	Ist and IInd semester
10	Forensic Science & Cyber Security (Single Major)	Ist and IInd semester

This is effective from the Academic Year 2024-25 and onwards.

All concerned are requested to note the contents of this circular and bring the notice to the students, teachers and staff for their information and necessary action.

University Campus,
Chhatrapati Sambhajinagar
-431 004.
Ref.No. SU/Sci./2024/27121-28
Date:-27.05.2024.

★
★
★
★
★
★★★★★
★

Deputy Registrar,
Academic Section.

..2..

Copy forwarded with compliments to :-

1] **The Principal of all concerned Colleges,**
Dr. Babasaheb Ambedkar Marathwada University,

2] **The Director, University Network & Information Centre, UNIC, with a request to upload this Circular on University Website.**

Copy to :-

1] The Director, Board of Examinations & Evaluation, Dr.Babasaheb Ambedkar Marathwada University,Chhatrapati Sambhajinagar.

2] The Section Officer,[B.Sc.Unit] Examination Branch, Dr.Babasaheb Ambedkar Marathwada University,Chhatrapati Sambhajinagar.

3] The Programmer [Computer Unit-1] Examinations, Dr.Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar.

4] The Programmer [Computer Unit-2] Examinations, Dr.Babasaheb Ambedkar Marathwada University,Chhatrapati Sambhajinagar.

5] The In-charge,[E-Suvidha Kendra], Rajarshi Shahu Maharaj Pariksha Bhavan, Dr.Babasaheb Ambedkar Marathwada University,Chhatrapati Sambhajinagar.

6] The Public Relation Officer, Dr.Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar.

7] The Record Keeper, Dr.Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar.

Dr. Babasaheb Ambedkar Marathwada University

Chhatrapati Sambhajinagar- 431001

B.C.A (Science) Degree Programme

(Three Year / Four Years (Hons) /Four Years (Hons with Research)

Course Structure and syllabus for BCA 1st year

(Revised)

(AS PER NEP-2020)

Major: Bachelor of Computer Application (Science)
(Single Major)

Effective from 2024-25

PREFACE

As we stand on the threshold of a new era in education, the dawn of the National Education Policy 2020 illuminates our path toward a holistic, inclusive, and progressive educational landscape. The Bachelor of Computer Application (B. C. A.) curriculum outlined herein reflects the ethos and aspirations of this transformative policy, aiming to equip learners with the knowledge, skills, and values necessary to thrive in the dynamic world of the 21st century.

At its core, the National Education Policy 2020 envisions an educational framework that is learner-centric, multidisciplinary, and geared towards fostering creativity, critical thinking, and innovation. It emphasizes the integration of knowledge across disciplines, breaking down traditional silos to encourage holistic understanding and application of concepts. The Bachelor of Computer Application (B. C. A.) curriculum embodies these principles by offering a diverse array of courses spanning various scientific domains, while also incorporating interdisciplinary studies to nurture well-rounded graduates capable of addressing complex challenges with agility and insight.

Furthermore, the curriculum is designed to promote experiential learning, research, and hands-on exploration, recognizing the importance of practical engagement in deepening understanding and cultivating real-world skills. Through laboratory work, field experiences, internships, and project-based learning opportunities, students will have the chance to apply theoretical knowledge in practical settings, develop problem-solving abilities, and cultivate a spirit of inquiry and discovery.

Integral to the National Education Policy 2020 is the commitment to inclusivity, equity, and access to quality education for all. The Bachelor of Computer Application (B. C. A.) curriculum reflects this commitment by embracing diversity in perspectives, backgrounds, and experiences, and by fostering an inclusive learning environment where every student feels valued, supported, and empowered to succeed.

Moreover, the curriculum emphasizes the cultivation of ethical values, social responsibility, and global citizenship, instilling in students a sense of accountability towards society and the environment. By integrating courses on ethics, sustainability, and social sciences, the Bachelor of Computer Application (B. C. A.) program aims to produce graduates who are not only proficient in their respective fields but also compassionate, ethical leaders committed to making a positive impact on the world.

As we embark on this journey of educational transformation guided by the National Education Policy 2020, the Bachelor of Computer Application (B. C. A.) curriculum stands as a testament to our collective vision of a more equitable, inclusive, and enlightened society. It is our hope that through rigorous academics, innovative pedagogy, and unwavering dedication to excellence, we can inspire the next generation of scientists, scholars, and change-makers to realize their full potential and contribute meaningfully to the advancement of knowledge and the betterment of humanity.

Introduction to Undergraduate Bachelor of Computer Application (Three Years Degree) (Four Year Honours) Programme

The Undergraduate Bachelor of Computer Application (BCA) program, whether as a standard three-year degree or an extended four-year honours degree, embodies the transformative aspirations outlined in the National Education Policy 2020 (NEP 2020). This comprehensive program serves as a cornerstone for nurturing a new generation of tech-savvy professionals equipped with the requisite knowledge and skills to thrive in the digital era. Aligned with the NEP 2020's emphasis on holistic and multidisciplinary education, the BCA curriculum integrates foundational concepts in computer science with interdisciplinary subjects, fostering a well-rounded educational experience. Furthermore, the program prioritizes experiential learning, practical application, and industry collaboration to ensure graduates are not only academically proficient but also industry-ready upon completion of their studies. By offering specialized tracks, elective courses, and research opportunities, the BCA program empowers students to tailor their learning journey according to their interests and career aspirations. Through this dynamic and forward-thinking approach, the BCA program under NEP 2020 not only equips students with technical expertise but also instills essential competencies such as critical thinking, problem-solving, and adaptability, thereby preparing them to become leaders and innovators in the rapidly evolving field of computer applications.

I. Courses of Study

The nature of courses within the Undergraduate Bachelor of Computer Application (BCA) program, whether in the standard three-year degree or the extended four-year honors degree, embodies a blend of theoretical foundations and practical applications. Courses are meticulously designed to provide students with a solid grounding in core computer science principles, including programming languages, data structures, algorithms, and software engineering methodologies. Additionally, specialized courses delve into diverse areas such as database management, networking, web development, artificial intelligence, and cybersecurity, catering to the evolving needs of the digital landscape. Embracing the principles of flexibility and adaptability advocated by the NEP 2020, the BCA program offers a wide range of elective courses and interdisciplinary options, allowing students to explore niche areas of interest or complement their technical expertise with knowledge from other domains. Moreover, the program integrates hands-on projects, internships, and industry collaborations to bridge the gap between academia and the real-world application of knowledge, fostering practical skills, professional competencies, and entrepreneurial mindsets among students. In essence, the nature of courses within the BCA program reflects a dynamic and progressive approach, poised to equip students with the requisite knowledge, skills, and mindset to thrive in a digitally driven world.

II. Broad Categories of Courses

Within the Undergraduate Bachelor of Computer Application (BCA) program, the curriculum is structured to offer a comprehensive array of courses aimed at honing students' expertise in core subjects, cultivating proficiency in minor domains, enhancing skill sets, fostering vocational skill development, and imparting value-based education.

1. **Major Courses:** The Major subjects encompass foundational areas such as programming languages, software engineering, database management, computer networks, and operating systems, providing students with a robust understanding of fundamental concepts in computer science.
2. **Minor Courses:** Complementing these core subjects, Minor domains offer students the opportunity to delve deeper into specialized areas such as cybersecurity, artificial intelligence, data analytics, web development, or mobile applications, allowing them to develop niche expertise aligned with their career aspirations.
3. **Skill Enhancement Courses:** Skill enhancement courses are integrated throughout the curriculum to bolster students' technical prowess, critical thinking abilities, problem-solving skills, and communication proficiency, ensuring they are well-equipped to tackle real-world challenges.
4. **Vocational Skill Courses**: Vocational skill development initiatives are woven into the fabric of the

program, offering practical training in areas such as project management, teamwork, leadership, and entrepreneurship, empowering students to excel in diverse professional settings.

5. **Value Education Courses**: Concurrently, value courses underscore the importance of holistic development by instilling ethical values, social responsibility, cultural sensitivity, and environmental awareness among students, nurturing them into well-rounded individuals with a global outlook.
6. **Ability Enhancement Courses**: Additionally, Ability Enhancement Courses provide avenues for students to explore interdisciplinary subjects, develop interdisciplinary perspectives, and enhance their employability through interdisciplinary learning experiences.
7. **Generic Elective/Open Elective Courses**: Flexible courses allowing students to explore diverse interdisciplinary subjects beyond their core discipline.
8. **Discipline Specific Elective Courses**: Specialized courses offering in-depth knowledge and expertise in specific areas within the student's chosen field of study.
9. **Indian Knowledge Courses**: Courses providing insight into India's cultural heritage, history, philosophy, and traditions, fostering appreciation and understanding of its diverse heritage.
10. **Co-Curricular Courses**: Courses enhancing students' holistic development through extracurricular activities, practical experiences, and skill-building initiatives, promoting personal growth and employability.

Collectively, these elements within the BCA program ensure that graduates emerge not only as technically proficient professionals but also as socially responsible leaders poised to make meaningful contributions to society.

Programme Educational Objectives (PEOs):

These Programme Educational Objectives delineate the core principles guiding the Bachelor of Computer Applications curriculum, underscoring our commitment to nurturing graduates equipped to excel in their careers, contribute meaningfully to society, and lead purposeful lives amidst the rapid evolution of technology.

1. **Mastery of Discipline-Specific Knowledge**: Graduates of the Bachelor of Computer Applications program will demonstrate a profound comprehension of fundamental principles, theories, and methodologies in computer application development and related fields. This expertise will empower them to dissect intricate computational problems, devise innovative solutions, and contribute to advancements in the realm of computer applications.
2. **Interdisciplinary Proficiency**: Graduates will possess the adeptness to synthesize knowledge and skills from various domains within computer science, fostering a holistic approach to problem-solving and innovation. They will be equipped to tackle multifaceted challenges by integrating diverse perspectives and methodologies, ensuring comprehensive solutions in the ever-evolving landscape of technology.
3. **Critical Thinking and Analytical Skills**: Graduates will cultivate robust critical thinking abilities, enabling them to scrutinize information rigorously, analyze data effectively, and make well-founded decisions grounded in evidence. Proficiency in logical reasoning and scientific methodologies will empower them to address complex computational problems and spearhead novel solutions.
4. **Leadership and Innovation**: Graduates will exhibit leadership qualities and an entrepreneurial mindset essential for catalyzing positive change in technological spheres. They will showcase creativity, resilience, and adaptability, leveraging innovation to confront intricate challenges and capitalize on opportunities for advancement within the dynamic landscape of computer applications.
5. **Global Citizenship and Cultural Sensitivity:** Graduates will embrace a global perspective and cultural sensitivity, acknowledging the interconnectedness of diverse communities in the digital age. They will actively engage in cross-cultural dialogue, embrace diversity, and contribute to the enrichment of knowledge and understanding on a global scale, fostering collaboration and cooperation across borders.

Programme Outcomes (POs)

These outcomes are designed to equip graduates with the technical expertise, analytical acumen, ethical sensibilities, and lifelong learning capabilities nccessary to thrive in the dynamic landscape of computer application.

1. **PO 1 – Disciplinary knowledge:**
   Graduates will adeptly apply mathematical principles, algorithmic paradigms, and core computing fundamentals in the modeling, design, and development of computer-based systems, leveraging advanced technologies to address contemporary challenges.
2. **PO 2 – Scientific reasoning/ Problem analysis:**
   Graduates will demonstrate advanced analytical skills to systematically analyze, categorize, and formulate solutions for multifaceted problems encountered within the domain of computer applications, utilizing cutting-edge technologies to enhance problem-solving capabilities.
3. **PO 3 – Problem solving:**
   Graduates will engineer software solutions to address complex scientific, business, and societal challenges, integrating considerations for modern technologies while prioritizing public health, safety, and environmental sustainability.
4. **PO 4 – Environment and sustainability:**
   Graduates will comprehend the environmental and societal impact of software solutions, striving to develop sustainable applications that promote societal well-being within the context of modern technological advancements.
5. **PO 5 – Modern tool usage:**
   Graduates will proficiently utilize contemporary software development tools and methodologies to facilitate efficient and collaborative development practices, incorporating emerging technologies seamlessly into their workflows.
6. **PO 6 – Ethics:**
   Graduates will navigate ethical complexities inherent in computer application development, upholding professional integrity and social responsibility within the dynamic landscape of technology integration.
7. **PO 7 – Cooperation / Teamwork:**
   Graduates will collaborate effectively as integral members or leaders of interdisciplinary teams, leveraging diverse skill sets and perspectives to achieve collective objectives in software development projects.
8. **PO 8 – Communication Skills:**
   Graduates will demonstrate proficiency in communicating technical concepts and insights to diverse audiences, adeptly preparing and presenting technical documentation tailored to the needs of stakeholders in computer application projects.
9. **PO 9 – Self-directed and Life-long Learning:**
   Graduates will exhibit a proactive commitment to continuous self-improvement and professional development, recognizing the imperative of lifelong learning to remain abreast of evolving technologies and industry trends.
10. **PO 10 – Enhance the research culture and uphold scientific integrity and objectivity:**
    Graduates will actively contribute to fostering a vibrant research culture, upholding the principles of scientific integrity, objectivity, and reproducibility in their scholarly pursuits within the diverse realms of computer applications.

Programme Specific Outcomes (PSOs):

1. **PSO1. Software Application Development Excellence**: Apply programming paradigms and software engineering principles, practices, and tools to analyze, design, implement, test, and maintain software systems that meet quality standards and user requirements.
2. **PSO2. Web Application Development Mastery**: Design and develop dynamic and interactive web applications using modern web technologies and frameworks, ensuring compatibility, performance, and security across different platforms and devices.
3. **PSO3. Data-driven Decision Making:** Utilize data analysis techniques, statistical models, and visualization tools to analyze, interpret, and present data effectively for decision-making and problem-solving in diverse domains.
4. **PSO4. Cybersecurity Implementation:** Implement security measures, conduct risk assessments, and respond to security incidents to protect information assets and mitigate cyber threats effectively, ensuring data confidentiality, integrity, and availability.
5. **PSO5. Cloud Computing Integration:** Design, deploy, and manage scalable and cost-effective cloud-based solutions using cloud computing technologies and platforms, ensuring reliability, availability, and performance to meet business needs and enable digital innovation.
6. **PSO6. User Experience Innovation:** Design and evaluate user interfaces and interactive systems using human-centered design approaches to create engaging, accessible, and usable experiences that meet user needs and preferences.

Structure of B. C. A (Science) (Three / Four Years Honours / Honours with Research Degree) Programme with Multiple Entry and Exit Options

Subject (Major): Computer Application

BSc First Year: 1st Semester

Course Type	Course Code	Course Name	Teaching Scheme (Hrs / Week)		Credits Assigned		Total Credits
			Theory	Practical	Theory	Practical	
Major (Core) M1 Mandatory	DSC-1	Fundamentals of Computer Application	2		2		2+2 = 4
	DSC-2	Practical based on DSC-1		4		2	
Major (Core) M2 Mandatory	DSC-3	C-Programing-I	2		2		2+2 = 4
	DSC-4	Practical based on DSC-3		4		2	
Major (Core) M3 Mandatory	DSC-5	Operating System-I	2		2		2+2 = 4
	DSC-6	Practical based on DSC-5		4		2	
Generic / Open Elective (GE/OE) (Choose any one from pool of courses) **It should be chosen compulsorily from the faculty other than that of Major**	GE/OE-1	To be chosen from other faculty	2		2		2
SEC (Skill Enhancement Courses) (Choose any one from pool of courses)	SEC-1	SEC-1A Animation Designing or SEC-1B Graphics Designing	1		1		2
	SEC-2	Practical based on SEC-1		2		1	
AEC, VEC, IKS	AEC-1	English (Common for all the faculty)	2		2		2+2 =4
	IKS-1	Choose any one from pool of courses	2		2		
OJT/ FP/CEP/CC/RP	CC-1	Health and Wellness (Common for all the faculty)		4		2	2
			13	**18**	**13**	**09**	**22**

GE/OE-: **Cyber Security** (This course will be available for the students from other faculty)

BSc First Year: 2nd Semester

Course Type	Course Code	Course Name	Teaching Scheme (Hrs / Week)		Credits Assigned		Total Credits
			Theory	Practical	Theory	Practical	
Major (Core) M1 Mandatory	DSC-7	Data Structures-I	2		2		2+2 = 4
	DSC-8	Practical based on DSC-7		4		2	
Major (Core) M2 Mandatory	DSC-9	C-Programming-II	2		2		2+2 = 4
	DSC-10	Practical based on DSC-9		4		2	
Major (Core) M3 Mandatory	DSC-11	Operating System-II	2		2		2+2 = 4
	DSC-12	Practical based on DSC-11		4		2	
Generic / Open Elective (GE/OE) (Choose any one from pool of courses) **It should be chosen compulsorily from the faculty other than that of Major**	GE/OE-2	To be chosen from other faculty	2		2		2
VSC (Vocational Skill Courses) (Choose any one from pool of courses)	VSC-1	VSC-1A PC-Maintenance-1 or VSC-1B Office Tools-I	1		1		2
	VSC-2	Practical based on VSC-1		2		1	
AEC, VEC, IKS	AEC-1	English (Common for all the faculty)	2		2		2+2 =4
	VEC-1	Constitution of India (Common for all the faculty)	2		2		
OJT/ FP/CEP/CC/RP	CC-2	Yoga Education / Sports and Fitness (Common for all the faculty)		4		2	2
			13	**18**	**13**	**09**	**22**
Exit Option: Award of UG Certificate in 3 Majors with 44 credits and an additional 4 credits of core NSQF course / Internship OR continue with Major and Minor							

GE/OE-2: **E-Commerce** (This course will be available for the students from other faculty)

Detailed Illustration of Courses included in 1st and 2nd semester:

1) **Major** (Core) subjects are mandatory.

DSC-1 : This is a 2 credit theory course corresponding to Major (core) subject M1
DSC-2 : This is a 2 credit practical course based on DSC-1
DSC-3 : This is a 2 credit theory course corresponding to Major (core) subject M2
DSC-4 : This is a 2 credit practical course based on DSC-3
DSC-5 : This is a 2 credit theory course corresponding to Major (core) subject M3
DSC-6 : This is a 2 credit practical course based on DSC-5

2) **Generic / Open Elective (GE/OE)**: (Needs to be chosen (any one) from pool of courses available at respective college). **These courses should be chosen compulsorily from faculty other than that of Major.**

   a. GE/OE -1 : This is a 2 credit theory course should be chosen compulsorily from faculty other than that of Major. (Sem-I)
   b. GE/OE -2 : This is a 2 credit theory course should be chosen compulsorily from faculty other than that of Major. (Sem-II)

3) **SEC** (Skill Enhancement Courses) : Choose any one from pool of courses. These courses needs to be designed to enhance the technical skills of the students in specific area.
   a. **SEC-1 :** This is a 1 credit theory course to enhance the technical skills of the students in specific area.
   b. **SEC**-2 : This is a 1 credit practical course based on SEC-1.

4) **VSC** (Vocational Skill Courses) : Choose any one from pool of courses. These courses should be based on Hands on Training corresponding to Major (core) subject.
   c. **VSC-1 :** This is a 1 credit theory course based Hands on Training corresponding to Major (core) subject.
   d. **VSC**-2 : This is a 1 credit practical course based on VSC-1

5) **AEC** (Ability Enhancement courses): The focus of these courses should be based on linguistic and communication skills.

AEC-1 : English

This is a 2 credit theory course based on linguistic proficiency. It will be common for all the faculty.

AEC-2 : English

This is a 2 credit theory course based on linguistic proficiency. It will be common for all the faculty.

1) **IKS** (Indian Knowledge System) : The courses related to traditional and ancient culture of India will be included in this section. The respective college will have to choose one of the courses from the pool of courses designed by the University.

IKS-1 : To be chosen from the pool of courses designed by the University
This is a 2 credit theory course based on Indian Knowledge System. It will be common for all the faculty

2) **VEC** (Value Education Courses): The courses such as understanding India, Environmental Science / Education, Digital and Technological solutions etc will be part of Value Education Courses.

VEC-1 : Constitution of India
This is a 2 credit theory course based on value education. It will be common for all the faculty

3) **CC** (Curricular Courses): The courses such as Health and wellness, Yoga education, Sports and Fitness, Cultural activities, NSS/NCC, Preforming Arts.

CC-1 : Health and Wellness
This is a 2 credit practical course based on Co-curricular activities. It will be common for all the faculty

CC-2 : Yoga education / Sports and Fitness
This is a 2 credit practical course based on Co-curricular activities. It will be common for all the faculty

Programme Educational Objectives (PEOs) :

Programme Educational Objectives (PEOs) for the Bachelor of Science Curriculum under the National Education Policy 2020:

1. **Mastery of Discipline-Specific Knowledge**: Graduates of the Bachelor of Science program will demonstrate a deep understanding of fundamental principles, theories, and methodologies in their chosen scientific discipline, enabling them to analyze complex problems, propose innovative solutions, and contribute to advancements in their field.

2. **Interdisciplinary Proficiency**: Graduates will possess the ability to integrate knowledge and skills from multiple scientific disciplines, fostering a holistic approach to problem-solving and innovation. They will be equipped to address multifaceted challenges by drawing upon diverse perspectives and methodologies.

3. **Critical Thinking and Analytical Skills:** Graduates will develop strong critical thinking abilities, enabling them to evaluate information rigorously, analyze data effectively, and make informed decisions based on evidence. They will demonstrate proficiency in applying logical reasoning and scientific methods to solve problems and generate new knowledge.

4. **Leadership and Innovation:** Graduates will demonstrate leadership qualities and entrepreneurial mindset, capable of initiating and driving positive change in their organizations and communities. They will exhibit creativity, resilience, and adaptability, harnessing innovation to address complex challenges and seize opportunities for growth and advancement.

5. **Global Citizenship and Cultural Sensitivity**: Graduates will possess a global perspective and cultural sensitivity, recognizing the interconnectedness of diverse communities and the importance of collaboration across borders. They will engage in cross-cultural dialogue, embrace diversity, and contribute to the advancement of knowledge and understanding on a global scale.

These Programme Educational Objectives serve as guiding principles for the Bachelor of Science curriculum, reflecting our commitment to nurturing well-rounded graduates who are prepared to excel in their careers, contribute to society, and lead meaningful lives in a rapidly changing world.

Programme Outcomes (POs) :

The National Education Policy (NEP) 2020 for India emphasizes several key aspects for Bachelor of Science (B.Sc.) programs, aiming to produce graduates who are not only well-versed in their respective disciplines but also equipped with skills necessary for holistic development and employability. While specific program outcomes may vary between institutions and disciplines within B.Sc. programs, here are some common outcomes aligned with NEP 2020:

- **PO1. The citizenship and society:** Apply broad understanding of ethical and professional skill in science subjects in the context of global, economic, environmental and societal realities while encompassing relevant contemporary issues.

- **PO2. Environment and sustainability**: Apply broad understanding of impact of science subjects in a global, economic, environmental and societal context and demonstrate the knowledge of, and need for sustainable development.

- **PO3. Ethics:** Apply ability to develop sustainable practical solutions for science subject related problems within positive professional and ethical boundaries.

- **PO4. Individual and team work:** Function effectively as a leader and as well as team member in diverse/ multidisciplinary environments.

- **PO5. Communication:** Communicate effectively on complex science subject related activities with the scientific community in particular and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

- **PO6. Project management and finance**: Demonstrate knowledge and understanding of the first principles of science and apply these to one's own work as a member and leader in a team, to complete project in any environment.

- **PO7. Life-long learning**: Recognize the need for lifelong learning and have the ability to engage in independent and life-long learning in the broadest context of technological change.

These program outcomes align with the broader goals of NEP 2020 to transform higher education in India and prepare students for the challenges and opportunities of the 21st century. Board of Studies designing B.Sc. curricula are encouraged to incorporate these outcomes into their program objectives and learning outcomes.

Programme Specific Outcomes (PSOs):

(Programme specific outcomes are discipline / major specific. Different major will have different PSOs. **Following is the example of PSOs for Electronics Major. Respective BoS is expected to draft PSOs related to their Major)**

PSO1. **Domain knowledge:** Apply the knowledge of electronics fundamental, and advanced areas of Electronics to provide comprehensive solution of problems in complex electronics.

PSO2. **Problem Analysis:** Identify electronics related problems at varied complexity and analyze the same to formulate/ develop substantiated conclusion using first principles of Electronics

PSO3. **Design Development of solutions:** Design/ develop solutions for problems at varied complexity in various areas of Electronics to address changing challenges put forward by market demand/ stakeholder

PSO4. **Conduct Investigation of complex problems:** Use established knowledge and methods to design of experiments, analyze resulting data and interpret the same to provide valid conclusions.

PSO5. **Modern tools:** Create, select, and apply appropriate techniques, resources, and modern electronics and relevant IT tools including prediction and modeling to complex electronics technology related activities with clear understanding of the limitations.

B.C.A.

Semester - I

DSC-1: Fundamentals of Computer Application

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Perquisite: There are no perquisites for this course

Learning Objectives of the Course:

i) Ability to identify and explain computer system components, including hardware and software.
ii) To be able to demonstrate proficiency in utilizing word processing, spreadsheet, presentation, and database management software.
iii) To be able to apply fundamental programming concepts using beginner-friendly languages to create simple programs.
iv) To understand and analyze the ethical, social, and economic implications of computer applications to be able to grasp their role in shaping society.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Demonstrate a thorough understanding of computer hardware and software components, their functions, and interactions within a computing system.
ii) Effectively utilize a range of productivity software applications to create, edit, and manage documents, data, and presentations.
iii) Apply fundamental programming concepts using beginner-friendly languages to develop simple programs for solving basic computational problems.
iv) Analyze and assess the ethical, social, and economic implications of computer applications, including cybersecurity, data privacy, and emerging technologies.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module 1: Introduction to Computer Fundamentals** • **Introduction to Computers and Their Evolution** Definition and historical background of computers, Evolution from early computing devices to modern computers • **Computer Hardware Components** Central Processing Unit (CPU), Memory (RAM), Storage Devices (Hard Disk Drive, Solid State Drive), Input Devices (Keyboard, Mouse, Touchpad), Output Devices (Monitor, Printer) • **Operating Systems** Functions of an operating system, Types of operating systems (Windows, macOS, Linux) • **Computer Software** System Software (Operating Systems), Application Software (Word Processors, Spreadsheets, Presentation Software) • **Computer Networks** Basics of Networking, Introduction to the Internet and World Wide Web • **Data Representation** Binary System and its importance in computing, ASCII and Unicode for character encoding	**10 Hrs**

II	**Module 2: Software Applications and Productivity Tools** • **Word Processing Software** Creating and Formatting Documents (Microsoft Word, Google Docs) • **Spreadsheet Software** Data Entry, Formulas, Functions, Charts (Microsoft Excel, Google Sheets) • **Presentation Software** Creating Slideshows, Adding Multimedia Elements (Microsoft PowerPoint, Google Slides) • **Database Management Systems** Introduction to Databases, Data Manipulation (Microsoft Access, MySQL) • **Graphics and Multimedia Applications** Image Editing, Audio/Video Playback (Adobe Photoshop, Audacity) • **Introduction to Programming Concepts** Understanding Algorithms, Variables, Control Structures (Scratch, Python)	**10 Hrs**
III	**Module 3: Computer Applications in Everyday Life and Work** • **Computer Ethics, Security, and Privacy** Ethical considerations in computing, Importance of cybersecurity and privacy protection • **Introduction to E-commerce and Online Transactions** Understanding e-commerce platforms, online payment systems (Payment Gateways) • **Office Automation Tools** Email, Calendars, Document Collaboration (Microsoft Outlook, Google Workspace) • **Cloud Computing** Concepts, Applications, Benefits (Google Drive, Dropbox) • **Mobile Computing** Smartphones, Apps, Mobile Platforms (iOS, Android) • **Emerging Trends and Future of Computing** Artificial Intelligence (AI), Internet of Things (IoT), Blockchain	**10 Hrs**

Text Books:

1. Miller, Michael. "Computer Basics Absolute Beginner's Guide, Windows 10 Edition", Pearson, 2021. [Module 1: Likely covers chapters on Introduction to Computers, Hardware Components, Software Applications, and Basic Troubleshooting.]
2. O'Leary, Timothy, Linda I. O'Leary, and Daniel O'Leary. "Computing Essentials 2022", McGraw-Hill Education, 2022. [Module 2: Likely covers chapters on Computer Hardware, Software, Networks, Internet, and Productivity Software Applications.]
3. Jukic, Nenad, Susan Vrbsky, and Svetlozar Nestorov. "Database Systems: Introduction to Databases and Data Warehouses", Pearson, 2018. [Module 2: Likely covers chapters on Introduction to Databases, Data Modeling, Database Design, Implementation, and Management.]
4. Dale, Nell, and John Lewis. "Computer Science Illuminated", Jones & Bartlett Learning, 2018. [Module 3: Likely covers chapters on Introduction to Computer Science, Programming Concepts, Ethics, Security, and Emerging Technologies.]
5. Laudon, Kenneth C., and Carol Guercio Traver. "E-Commerce 2021", Pearson, 2021. [Module 3: Likely covers chapters on Introduction to E-commerce, Business Models, Technologies, Security Issues, and Emerging Trends.]

Reference Books:

1. Deitel, Paul J., and Harvey Deitel. "Java: How to Program (Early Objects)", Pearson, 2017. [Module 2: Introduction to Programming Concepts]
2. Tanenbaum, Andrew S., and David J. Wetherall. "Computer Networks", Pearson, 2018. [Module 2: Computer Networks]
3. Sebesta, Robert W. "Concepts of Programming Languages", Pearson, 2015. [Module 3: Programming Concepts]
4. Rosen, Kenneth H. "Discrete Mathematics and Its Applications", McGraw-Hill Education, 2018. [Module 3: Mathematical Foundations]
5. Stair, Ralph M., and George W. Reynolds. "Principles of Information Systems", Cengage Learning, 2020. [Module 3: Information Systems]

E-contents:

1. **Khan Academy**:
   - Course: Computer Science
   - Link: https://www.khanacademy.org/computing

DSC-2: Lab Course -I (Based on DCS-1)

Total Credits: 02
Maximum Marks : 50

Total Contact Hours: 60 Hrs

Lab Session No.	Lab Title/Topic
1	Aim: To familiarize students with various computer hardware components and their functions through hands-on identification and labeling exercises.
2	Aim: To guide students through the process of installing and setting up an operating system on a virtual machine or physical computer.
3	Aim: To introduce students to basic networking concepts and configurations by setting up a small peer-to-peer network and configuring network settings.
4	Aim: To help students understand the binary system and its importance in computing through practical exercises converting between decimal and binary numbers.
5	Aim: To teach students document creation and formatting skills using word processing software such as Microsoft Word or Google Docs.
6	Aim: To develop students' proficiency in spreadsheet software by guiding them through creating and using formulas and functions for data analysis and manipulation.
7	Aim: To enhance students' presentation skills by guiding them through the design and creation of effective slideshows using presentation software like Microsoft PowerPoint or Google Slides.
8	Aim: To introduce students to database concepts and basic data manipulation tasks using database management systems such as Microsoft Access or MySQL.
9	Aim: To familiarize students with image editing techniques using graphics editing software like Adobe Photoshop.
10	Aim: To introduce students to basic programming concepts such as algorithms, variables, and control structures through hands-on projects in Scratch.
11	Aim: To raise awareness among students about computer ethics, cybersecurity, and privacy protection through interactive discussions and case studies.
12	Aim: To provide students with hands-on experience in setting up and managing an e-commerce platform, including product listings, customer orders, and payments.
13	Aim: To raise awareness among students about computer ethics, cybersecurity, and privacy protection through interactive discussions and case studies.
14	Aim: To introduce students to cloud computing concepts and applications by exploring cloud storage and collaboration tools like Google Drive and Dropbox. Mobile App Development Basics Lab
15	Aim: To introduce students to mobile computing platforms and app development concepts, allowing them to create simple mobile apps using platforms like MIT App Inventor or Thunkable.

DSC-3: C- Programming-I

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Acquire foundational knowledge of C programming concepts, including syntax, data types, operators, and control structures.
ii) Develop problem-solving skills through algorithmic thinking and practice, enabling efficient solutions in the C programming language.
iii) Master program design and implementation in C, encompassing structure, data types, control flow, and input/output operations.
iv) Demonstrate proficiency in algorithm design, flowchart interpretation, and translating algorithms into executable code in C.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Develop, compile, execute, and debug C programs proficiently, thereby demonstrating their mastery of the programming language.
ii) Apply problem-solving techniques effectively to various programming tasks, including algorithm design, flowchart interpretation, and implementation in C.
iii) Understand program structure, control flow mechanisms, and data types in C, enabling them to design and implement efficient and structured programs.
iv) Demonstrate strong algorithmic thinking skills, allowing them to formulate and implement algorithms for solving a wide range of computational problems using the C programming language.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module I: Introduction to C Programming and Basic Problem-Solving Techniques** **Introduction:** An Overview of C, History of C language, C as a Structured Language, Features of C **Basic Elements & Operators:** Character set, C Token, Identifier & Keywords, Variables, Constant and its types (Integer constant, floating-point constant, character constant, string constants), Operators (Arithmetic, Relational, Logical, Unary operators: Increment & decrement, Assignment, Conditional operator), Precedence & Associativity of Operators **Algorithms in Programming:** Introduction to Algorithms, Characteristics of Good Algorithms, Algorithm Design Techniques, Common Algorithms and Problem-Solving Techniques, Examples and Practice Exercises **Flowcharts in Programming:** Introduction to Flowcharts, Symbols and Conventions used in Flowcharts, Flowchart Designing Techniques, Importance of Flowcharts in Programming, Examples and Practice Exercises	**10 Hrs**

II	**Module II: Data Types and Program Structure** **Data Types:** Data Types: int, char, float, double, Declaration & Initialization, Type modifiers: long, short, signed & unsigned. **C Program & I/O statements:** Structure of C Program, Compilation & Execution of C program, I/O Introduction, Formatted Input/Output functions: scanf & printf, Escape sequence characters, Library functions: General & Maths	**10 Hrs**
III	**Module III: Control Flow and Iterative Statements** **Control and Iterative Statements:** Simple if, nested if, if-else, else-if ladder, Switch-case statement, The conditional expression (? : operator), while and do-while loop, for loop, break & continue statement, goto statement	**10 Hrs**

Text Books:

1. Kanetkar, Y.P. "Let us C", bpb publication.
2. Balaburuswamy, E. "Programming in C", Tata Macgraw Hill.
3. Goterfried. "Programming in C", Shaums' Series.

Reference Books:

1. Kooper, Moolish. "Spirit of 'C'".

E-contents:

1. Websites like GeeksforGeeks (https://www.geeksforgeeks.org/) and TutorialsPoint (https://www.tutorialspoint.com/) offer comprehensive tutorials on flowcharts, algorithms, and data structures.
2. YouTube channels such as CS Dojo (https://www.youtube.com/user/CSDojo) and freeCodeCamp.org (https://www.youtube.com/channel/UC8butISFwT-Wl7EV0hUK0BQ) provide video tutorials on these topics.
3. **C Programming**:

- For learning C programming, you can check out free online courses on platforms like Coursera (https://www.coursera.org/), edX (https://www.edx.org/), and Udemy (https://www.udemy.com/).
- There are also numerous tutorials and resources available on websites like GeeksforGeeks, TutorialsPoint, and W3Schools (https://www.w3schools.com/).
- YouTube channels like Programming with Mosh (https://www.youtube.com/user/programmingwithmosh) and TheNewBoston (https://www.youtube.com/user/thenewboston) offer video tutorials on C programming.

DSC-4: Lab Course -II (Based on DCS-3)

Total Credits: 02
Maximum Marks : 50 Total Contact IIours: 60 IIrs

Lab Session No.	Lab Title/Topic
1	Write a C program to find the area of a circle. Start by drafting the algorithm to compute the area, visually represent the logic through a flowchart, and conclude by implementing the solution in C.
2	Develop a C program to calculate the average of three numbers. Begin by outlining the algorithm to compute the average, create a flowchart to illustrate the process, and finalize the solution in C.
3	Create a C program to find the maximum among two numbers using if-else statements. Begin by formulating the decision-making algorithm, visually represent the logic with a flowchart, and conclude by implementing the solution in C.
4	Write a C program to determine whether an entered number is even or odd using if-else statements. Start by outlining the decision-making process, create a flowchart to depict the logic, and finalize the solution in C.
5	Develop a C program to determine whether an entered number is positive, negative, or zero using else-if ladder. Begin by formulating the decision-making algorithm, visually represent the logic with a flowchart, and conclude by implementing the solution in C.
6	Design a C program to print the grade of a student based on the percentage using else-if ladder. Start by outlining the grading algorithm, create a flowchart to illustrate the grading process, and finalize the solution in C.
7	Create a C program to print the weekday based on a given number using switch-case statements. Begin by formulating the decision-making algorithm, visually represent the logic with a flowchart, and conclude by implementing the solution in C.
8	Write a C program to check whether an entered character is a vowel or not using switch-case statements. Start by outlining the decision-making process, create a flowchart to depict the logic, and finalize the solution in C.
9	Develop a C program to find the factorial of a given number using a while loop. Begin by outlining the iterative algorithm, visually represent the process with a flowchart, and conclude by implementing the solution in C.
10	Design a C program to find the sum of the first 10 natural numbers using a while loop. Start by formulating the iterative algorithm, create a flowchart to illustrate the process, and finalize the solution in C.
11	Create a C program to print odd numbers from 1 to N using a do-while loop. Begin by formulating the iterative algorithm, visually represent the process with a flowchart, and conclude by implementing the solution in C.
12	Develop a C program to print the Fibonacci series using a do-while loop. Start by outlining the iterative algorithm, create a flowchart to illustrate the sequence, and finalize the solution in C.
13	Create a C program to print the following series using a for loop: 1 2 3 4 5 6 Begin by formulating the iterative algorithm, visually represent the series with a

	flowchart, and conclude by implementing the solution in C.
14	Design a C program to print prime numbers from 1 to 100 using a for loop. Start by outlining the iterative algorithm, create a flowchart to illustrate the process, and finalize the solution in C.
15	Develop a C program to check if a given year is a leap year or not. Begin by formulating the decision-making algorithm, visually represent the logic with a flowchart, and conclude by implementing the solution in C.

DSC-5: Operating System-I

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Understand the foundational principles of operating systems, including their role as resource managers, basic functions, and evolutionary development.
ii) Master the concept of process management, distinguishing between processes and threads, and evaluating CPU scheduling algorithms for efficiency.
iii) Demonstrate proficiency in inter-process communication and synchronization, recognizing the necessity of synchronization, implementing critical sections and semaphores, and employing various communication methods effectively.
iv) Apply operating system concepts to analyze and compare different system architectures, optimize resource utilization through process management techniques, and ensure program correctness and efficiency through synchronization mechanisms.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Compare different types of operating systems, considering their architectural features, strengths, and limitations.
ii) Optimize resource utilization and system performance through effective application of process management techniques and CPU scheduling algorithms.
iii) Design and implement robust synchronization mechanisms to ensure program correctness and efficiency in concurrent computing environments.
iv) Make informed design decisions regarding operating system configurations and architectures, considering factors such as reliability, efficiency, and scalability.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module I**: **Introduction to Operating System**: Introduction to Software, including its definition and classification, overview of operating systems as system software, and basic understanding of programs and processes; Operating System Fundamentals, covering the role of an operating system as a resource manager, basic functions, characteristics of modern operating systems, and a brief overview of the evolution of operating systems.	**10 Hrs**
II	**Module II: Process Management**: Explanation of the concept of a process including process states and control blocks, introduction to threads; CPU Scheduling, discussing types of schedulers, their criteria, and fundamental scheduling algorithms such as First-Come First-Served (FCFS), Shortest Job First (SJF), Priority Scheduling, and Round Robin.	**10 Hrs**
III	**Module III: Inter-Process Communication and Synchronization**: Discussion on the need for synchronization including concurrent and dependent processes, introduction to critical sections and semaphores, overview of inter-process communication methods, and basic understanding of synchronization problems	**10 Hrs**

	and solutions.	

Text Books:

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. "Operating System Concepts." Wiley, 2021

Reference Books:

1. Tanenbaum, Andrew S., and Herbert Bos. "Modern Operating Systems." Pearson, 2020.
2. Galvin, Peter Baer, et al. "Operating System Concepts Essentials." Wiley, 2022.
3. Stallings, William. "Operating Systems: Internals and Design Principles." Pearson, 2018.
4. Singh, A. K., and Shweta Singh. "Operating System." Pearson, 2019.

E-contents:

1. **Operating System Concepts Online Exercises**: A collection of interactive exercises and quizzes based on the "Operating System Concepts" textbook. Available at https://www.os-book.com/OS10/simulation/.
2. **Operating Systems: Three Easy Pieces**: An online book by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, covering fundamental concepts in operating systems. Available for free at http://pages.cs.wisc.edu/~remzi/OSTEP/.
3. **Operating System Lecture Notes**: Lecture notes and slides on operating system concepts from various universities, available through sites like MIT OpenCourseWare, Stanford Online, and Coursera.
4. **GeeksforGeeks - Operating System**: GeeksforGeeks offers a comprehensive collection of articles, tutorials, and coding exercises on operating systems concepts. Available at https://www.geeksforgeeks.org/operating-systems/.
5. **OSDev Wiki**: A wiki dedicated to operating system development, providing tutorials, articles, and resources for those interested in creating their own operating systems. Available at https://wiki.osdev.org/Main_Page..

DSC-6: Lab Course -III (Based on DCS-5)

Total Credits: 02
Maximum Marks : 50

Total Contact Hours: 60 Hrs

Lab Session No.	Lab Title/Topic
1	**Lab Session 1: Exploring Control Panel (Windows):** Objective: Familiarize yourself with the various system settings and configurations available in the Control Panel. 1. Question: How would you navigate through the Control Panel in Windows? 2. Task: Open Control Panel on your Windows system and explore different categories, such as System and Security, Network and Internet, and Programs.
2	**Lab Session 2: Understanding Task Manager (Windows):** Objective: Learn how to analyze resource usage, troubleshoot application issues, and manage processes using Task Manager. 1. Question: How can you use Task Manager to monitor system performance and manage processes in Windows? 2. Task: Open Task Manager on your Windows system and navigate through its tabs, such as Processes, Performance, and Startup.
3	**Lab Session 3: Exploring System Settings (Windows):** Objective: Understand how to customize system settings to suit your preferences and requirements. 1. Question: What system settings can be customized in Windows, and how can you access them? 2. Task: Open the System Settings on your Windows system and explore options for display settings, sound settings, power options, and accessibility features.
4	**Lab Session 4: Introduction to Command Prompt (Windows):** Objective: Gain hands-on experience with command-line operations and understand the role of Command Prompt in Windows. 1. Question: How do you access and use Command Prompt for command-line operations in Windows? 2. Task: Open Command Prompt on your Windows system and practice using basic commands such as cd, dir, mkdir, and del.
5	**Lab Session 5: Batch Scripting Basics (Windows):** Objective: Learn the syntax and structure of batch scripts and understand their role in automating repetitive tasks in Windows. 1. Question: What is batch scripting, and how can it be used to automate tasks in Windows? 2. Task: Create a simple batch script to perform file management tasks, such as copying files from one location to another.
6	**Lab Session 6: File Management Commands (DOS):** Objective: Gain proficiency in file management tasks using command-line operations in DOS. 1. Question: What are the basic file management commands in DOS, and how do you use them?

	2. Task: Practice using DOS commands such as copy, move, delete, and rename to manipulate files and directories.
7	**Lab Session 7: Disk Management Commands (DOS):** Objective: Understand disk management concepts and learn how to perform disk-related tasks in DOS environments. 1. Question: How can you manage disks and partitions using disk management commands in DOS? 2. Task: Use DOS commands such as format and chkdsk to format disks, check disk integrity, and manage disk partitions.
8	**Lab Session 8: System Configuration (DOS):** Objective: Learn how to customize system configuration settings to optimize system performance and functionality in DOS. 1. Question: What are the system configuration files in DOS, and how do you customize them? 2. Task: Edit the config.sys and autoexec.bat files to configure system startup settings and environment variables.
9	**Lab Session 9: First-Come First-Served (FCFS) Scheduling Algorithm:** Objective: Understand the basic principles of process scheduling and explore the FCFS algorithm through practical implementation. 1. Question: What is the FCFS scheduling algorithm, and how does it work? 2. Task: Implement the FCFS scheduling algorithm in a simple program and analyze its performance.
10	**Lab Session 10: Shortest Job First (SJF) Scheduling Algorithm:** Objective: Explore the advantages and limitations of the SJF algorithm in process scheduling. 1. Question: How does the SJF scheduling algorithm prioritize processes based on their burst times? 2. Task: Implement the SJF scheduling algorithm in a simple program and compare its performance with FCFS.
11	**Lab Session 11: Priority Scheduling Algorithm:** Objective: Understand how priority-based scheduling can be used to allocate resources and manage process execution. 1. Question: What is the priority scheduling algorithm, and how does it assign priorities to processes? 2. Task: Implement the priority scheduling algorithm in a simple program and evaluate its effectiveness in process management.
12	**Lab Session 12: Round Robin Scheduling Algorithm:** Objective: Explore the concept of time slicing and preemptive scheduling in Round Robin algorithm implementation. 1. Question: How does the Round Robin scheduling algorithm allocate CPU time to processes in a preemptive manner? 2. Task: Implement the Round Robin scheduling algorithm in a simple program and analyze its performance compared to other scheduling algorithms.
13	**Lab Session 13: Advanced Command Prompt Commands (Windows):** Objective: Gain proficiency in advanced command-line operations and system management tasks using Command Prompt in Windows. 1. Question: What are some advanced Command Prompt commands, and how can they be used for system administration tasks? 2. Task: Practice using advanced Command Prompt commands such as net, diskpart, and systeminfo to perform system administration tasks.

14	**Lab Session 14: Advanced Batch Scripting Techniques (Windows):** Objective: Learn advanced batch scripting concepts and techniques for creating robust and efficient automation scripts in Windows. 1. Question: How can you use advanced batch scripting techniques to automate complex tasks in Windows? 2. Task: Create a batch script that incorporates conditional statements, loops, and error handling to automate a multi-step process.
15	**Lab Session 15: System Optimization and Troubleshooting (Windows):** Objective: Apply the knowledge and skills acquired throughout the lab sessions to optimize system performance and address common issues in Windows environments. 1. Question: How can you optimize system performance and troubleshoot common issues in Windows? 2. Task: Use Task Manager, system settings, Command Prompt, and batch scripts to identify performance bottlenecks, optimize system settings, and troubleshoot system errors.

SEC-1A: Animation Designing

Total Credits: 01 Total Contact Hours: 15 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Understand Scratch's interface and features.
ii) Create basic motion animations and add sound effects.
iii) Learn costume and sprite animation techniques.
iv) Explore advanced animation controls and effects.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Navigate and utilize Scratch's interface effectively.
ii) Develop and present simple animated projects using Scratch.
iii) Apply basic animation principles to create engaging animations.
iv) Utilize advanced features in Scratch to enhance animation quality and interactivity.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	Module 1: Introduction to Scratch and Basic Animation Concepts Introduction to Scratch: Understanding the Interface Exploring Scratch's Sprite and Stage Features Understanding Motion Blocks for Basic Animation Introduction to Events and Control Blocks	**5 Hrs**
II	Module 2: Creating Basic Animations Creating Simple Motion Animations Adding Sound Effects and Background Music Introduction to Costume and Sprite Animation Using Control Blocks for Animation Timing Creating Interactive Animations with Events	**5 Hrs**
III	Module 3: Advanced Animation Techniques Using Variables for Animation Control Building Character Animations and Sequences Introduction to Pen and Drawing Effects for Animation Advanced Control Blocks for Animation Logic	**5 Hrs**

Reference Books:

1. Williams, Richard. "The Animator's Survival Kit." Faber & Faber, 2009.
2. Marji, Majed. "Programming Scratch: Learn to Program by Making Arcade Games." No Starch Press, 2013.
3. Sylvester, Tynan. "Designing Games: A Guide to Engineering Experiences." O'Reilly Media, 2013.
4. Shiffman, Daniel. "Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction." Morgan Kaufmann, 2015.

5. Rall, Hannes. "Animation: From Concept to Production." CRC Press, 2017.
6. Schell, Jesse. "The Art of Game Design: A Book of Lenses." CRC Press, 2008.
7. Ford Jr., Jerry Lee. "Scratch Programming for Teens." Cengage Learning, 2009.
8. Levy, David B. "Animation Development: From Pitch to Production." CRC Press, 2017.
9. Greenberg, Ira, Dianna Xu, and Deepak Kumar. "Creative Coding and Generative Art with Processing." Apress, 2013.
10. Nichols, Poppy, and Steve Roberts. "Exploring Animation Principles in Maya: Follow the Principles, Master the Art." CRC Press, 2019.

E-contents:

1. **Scratch Website:** The official Scratch website (scratch.mit.edu) provides comprehensive resources, tutorials, and project ideas for learning Scratch programming. It also hosts a vibrant community where users can share projects, collaborate, and seek help.
2. **YouTube Tutorials:** Many educators and enthusiasts create free tutorials on YouTube covering various aspects of Scratch programming and animation design. You can search for tutorials tailored to your level and interests.
3. **ScratchEd:** ScratchEd is an online community and resource hub for educators interested in teaching with Scratch. It offers free resources, curriculum guides, and professional development opportunities for educators.

SEC-2: Lab Course -I (Based on SEC-1A)

Total Credits: 01
Maximum Marks : 50

Total Contact Hours: 30 Hrs

Lab Session No.	Lab Title/Topic
1	**Lab 1: Introduction to Scratch Interface:** Objective: Understand the basic layout and features of Scratch for animation development. Task: Familiarize students with the Scratch interface, including the stage, sprites, and blocks palette.
2	**Lab 2: Exploring Sprite and Stage Features:** Objective: Gain proficiency in manipulating sprites and stage elements for animation creation. Task: Explore the capabilities of Scratch sprites and the stage, including resizing, rotating, and changing costumes.
3	**Lab 3: Motion Blocks for Basic Animation:** Objective: Understand the fundamentals of motion blocks and their application in animation. Task: Experiment with motion blocks to create basic animations, such as sprite movement and rotation.
4	**Lab 4: Introduction to Events and Control Blocks:** Objective: Understand how events and control blocks can be used to control animation sequences. Task: Learn how to use events and control blocks to trigger actions in Scratch animations.
5	**Lab 5: Creating Simple Motion Animations:** Objective: Develop proficiency in creating motion-based animations using Scratch. Task: Create simple motion animations using Scratch, incorporating basic motion blocks and sprite interactions.
6	**Lab 6: Adding Sound Effects and Background Music:** Objective: Explore the role of audio in enhancing the quality of animations. Task: Add sound effects and background music to Scratch animations using built-in sound blocks.
7	**Lab 7: Introduction to Costume and Sprite Animation:** Objective: Understand the concept of sprite animation and its application in creating engaging animations. Task: Experiment with costume changes and sprite animations to create dynamic visual effects.
8	**Lab 8: Using Control Blocks for Animation Timing:** Objective: Learn how to synchronize animation elements using control

	blocks. Task: Utilize control blocks to control the timing and sequence of animation events.
9	**Lab 9: Creating Interactive Animations with Events:** Objective: Explore interactive storytelling and game design concepts in animation creation. Task: Develop interactive animations using event blocks to respond to user input.
10	**Lab 10: Using Variables for Animation Control:** Objective: Understand the role of variables in dynamic animation control. Task: Implement variables to control animation parameters, such as speed and direction.
11	**Lab 11: Building Character Animations and Sequences:** Objective: Develop skills in character animation and storytelling through sequential animation. Task: Design and animate character sprites using multiple costumes and sequences.
12	**Lab 12: Introduction to Pen and Drawing Effects for Animation:** Objective: Explore the creative possibilities of pen and drawing effects in animation design. Task: Experiment with pen blocks and drawing effects to create custom animation elements.
13	**Lab 13: Advanced Control Blocks for Animation Logic:** Objective: Learn how to implement logic structures for advanced animation behaviors. Task: Use advanced control blocks, such as loops and conditionals, to create complex animation logic.
14	**Lab 14: Collaborative Animation Project:** Objective: Apply learned skills and techniques to a collaborative animation project. Task: Collaborate with peers to create a complex animation project incorporating various animation techniques and concepts.
15	**Lab 15: Animation Showcase and Reflection:** Objective: Evaluate animation projects, share feedback, and reflect on personal learning and growth. Task: Showcase completed animation projects to peers and reflect on the animation creation process.

SEC-1B: Graphics Designing

Total Credits: 01 Total Contact Hours: 15 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Understand Canva Interface: Familiarize yourself with the layout and tools available in Canva for creating graphics and designs.
ii) Create Visual Content: Learn to design various types of visual content such as social media posts, posters, and presentations using Canva's templates and design elements.
iii) Apply Design Principles: Gain knowledge of basic design principles such as layout, color theory, and typography, and apply them effectively in your Canva designs.
iv) Explore Collaboration and Sharing: Discover features for collaborating with others on design projects and learn how to share and export your designs for different purposes and platforms.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Design visually appealing graphics and layouts using Canva's tools and templates.
ii) Apply fundamental design principles such as layout, color theory, and typography to create professional-looking designs.
iii) Collaborate effectively with others on design projects using Canva's collaboration features.
iv) Share and export their designs for various purposes and platforms, demonstrating proficiency in using Canva for real-world applications.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module 1: Getting Started with Canva (5 hours)** 1. Introduction to Canva: Understanding the Interface (1 hour) 2. Exploring Canva's Design Tools and Features (2 hours) 3. Creating Your First Design: Poster or Social Media Graphic (2 hours)	**5 Hrs**
II	**Module 2: Design Principles and Techniques (5 hours)** 1. Understanding Design Principles: Composition, Color, Typography (1 hour) 2. Applying Design Techniques in Canva (2 hours) 3. Creating Visual Hierarchy and Balance in Designs (1 hour) 4. Incorporating Images and Illustrations Effectively (1 hour)	**5 Hrs**
III	**Module 3: Advanced Design Techniques (5 hours)** 1. Advanced Text Effects and Typography (1 hour) 2. Working with Layers and Grouping Objects (1 hour) 3. Using Canva Templates and Customizing Them (2 hours)	**5 Hrs**

	4. Designing Infographics and Presentations (1 hour)	

Reference Books:

E-contents:

1. **Canva Design School:** Canva offers a comprehensive Design School with tutorials, articles, and courses covering various design topics, from basic design principles to advanced techniques. You can access it here: https://www.canva.com/designschool/
2. **YouTube Tutorials:** Many content creators share tutorials and tips for using Canva on YouTube. You can find tutorials ranging from beginner to advanced levels, covering different aspects of design and Canva features.
3. **Canva's YouTube Channel:** Canva's official YouTube channel provides video tutorials, tips, and inspiration for using Canva effectively. You can find a variety of videos covering different design topics and techniques.
4. **Community Forums and Groups:** Joining online communities like Canva's Facebook groups or Reddit forums can be a great way to learn from others, ask questions, and get feedback on your designs.

SEC-2: Lab Course -I (Based on SEC-1B)

Total Credits: 01
Maximum Marks : 50

Total Contact Hours: 30 Hrs

Lab Session No.	Lab Title/Topic
1	**Lab 1: Introduction to Canva Interface** Objective: Understand the basic layout and features of Canva for design creation. Task: Familiarize students with the Canva interface, including tools, menus, and workspace organization.
2	**Lab 2: Exploring Canva's Design Tools and Features:** Objective: Gain proficiency in using various design tools and features in Canva. Task: Explore Canva's design tools such as text, shapes, backgrounds, and effects to create simple designs.
3	**Lab 3: Creating Your First Design: Poster or Social Media Graphic:** Objective: Apply basic design principles to create visually appealing posters or social media graphics. Task: Creating their first design project using Canva, focusing on layout, color, and content.
4	**Lab 4: Understanding Design Principles:** Objective: Learn fundamental design principles including composition, color theory, and typography. Task: Discuss and analyze examples of design compositions and typography, and their impact on visual communication.
5	**Lab 5: Applying Design Techniques in Canva:** Objective: Apply design techniques learned to create visually engaging designs in Canva. Task: Create designs in Canva using principles of balance, contrast, alignment, and proximity.
6	**Lab 6: Creating Visual Hierarchy and Balance in Designs:** Objective: Understand how to create visual hierarchy and balance in design compositions. Task: Design projects focusing on establishing visual hierarchy through font size, color contrast, and element placement.
7	**Lab 7: Incorporating Images and Illustrations Effectively:** Objective: Learn how to use images and illustrations to enhance design compositions. Task: Experiment with incorporating images and illustrations into design projects.
8	**Lab 8: Advanced Text Effects and Typography:** Objective: Explore advanced text effects and typography techniques in

	Canva. Task: Experiment with typography effects such as shadows, gradients, and text wrapping to create visually dynamic designs.
9	**Lab 9: Working with Layers and Grouping Objects:** Objective: Understand the concept of layers and object grouping for efficient design management. Task: Practice working with layers and grouping objects in Canva to organize design elements effectively.
10	**Lab 10: Using Canva Templates and Customizing Them:** Objective: Learn how to utilize Canva templates and customize them for specific design needs. Task: Explore Canva's template library, select a template, and customize it according to design requirements.
11	**Lab 11: Designing Infographics and Presentations:** Objective: Develop skills in creating infographics and presentations using Canva. Task: Design infographics and presentations in Canva, focusing on visual storytelling and information presentation.
12	**Lab 12: Advanced Design Projects:** Objective: Apply advanced design techniques learned to create complex design projects. Task: Work on advanced design projects such as branding materials, marketing collateral, or digital publications using Canva.
13	**Lab 13: Collaborative Design Project:** Objective: Collaborate with peers to create a design project, incorporating collective ideas and feedback. Task: Collaborate with classmates on a design project, sharing ideas, critiques, and contributions using Canva's collaborative features.
14	**Lab 14: Design Critique and Feedback Session:** Objective: Evaluate and provide constructive feedback on design projects created by peers. Task: Participate in a design critique session, offering feedback on peers' design projects and receiving feedback on your own.
15	**Lab 15: Portfolio Development and Presentation:** Objective: Compile and present a portfolio showcasing the design projects completed throughout the course. Task: Create a portfolio showcasing select design projects created in Canva, and present it to the class, highlighting design process, rationale, and outcomes.

This course will be available for the students from other faculty

GE/OE-1: Cyber Security

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

1. Make the student will understand Cyber Security, Data Privacy and Data Protection.
2. Students will acquainted with the Types of Security threats.
3. Make the student will understand Ethical Hacking, Email security: web authentication.

Course Outcomes (COs) :

After completion of the course, students will be able to –

1. Understands the concept and process of cyber security.
2. Understands the Online Dispute Resolution.
3. Knows the Network & Mobile Security Techniques

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	Cyber Security: Meaning and Scope Computer & Cyber Security: Types of Attacks, Types of Security threats, Hacking Techniques	**15 Hrs**
II	Database Security; Operating System Security 2. Advance Computers, Network & Mobile Security Techniques 3. Security issues: debit cards, credit cards, ATM, Secure Electronic Transactions	**15 Hrs**

Reference Books:

1. Information Security and Cyber Laws, by Pankaj Sharma. S.K. Kataria & Sons
2. Fundamentals of Cyber Security, by Bhushan, Rathore, Jamshed, BPB
3. Cyber-security for Beginners, by Raef Meeuwisse. Cyber Simplicity Ltd
4. A Handbook of E-commerce, by Nidhi Dhawan, Sun India Publications
5. E-Commerce in India: Economic and Legal Perspectives, Pralok Gupta, Sage Publications India Pvt. Ltd.

B.C.A.
Semester - II

DSC-7: Data Structures

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Prerequisites:

Basic understanding of C programs & arrays, hands on experience in decision making and looping constructs of C programming language will be a huge benefit.

Learning Objectives of the Course:

i) To provide fundamental knowledge of data structures and how they are organized/arranged in computer memory.
ii) To provide knowledge on how data structures are implemented and processed.
iii) To familiarize with basic techniques of algorithm analysis.
iv) To equip with the implementation techniques of complex algorithms of insertion, deletion and modification of data stored in various data structures.
v) To provide knowledge of the basic functioning of searching and sorting algorithms.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Understand and implement fundamental data structures like arrays, stacks, queues.
ii) Program data structures and use them effectively in solving computational problems.
iii) Evaluate basic algorithmic complexity to select suitable data structures and algorithms for problem-solving.
iv) Apply searching and sorting algorithms in practical scenarios with proper justifications.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module 1: Introduction to Data Structures, Algorithm Analysis and Arrays** Data Structures: Introduction to linear and non-linear data structures. Algorithm Analysis: Growth rates, Estimating the growth rate, Big O notation. Arrays: Need for Arrays, Linear Arrays, representation of linear arrays (row-major order, column-major order), Traversing, insertion, modification, deletion in linear array, merging linear arrays. 2-dimensional arrays introduction, representation of 2-dimensional array, sparse matrices.	**10 Hrs**
II	**Module 2: Stack, and Queue** Stack & Queue: Introduction, Operations on stack, stack implementation using arrays, Applications of Stack (Expression representation and evaluation), Expression notations (prefix, infix, postfix), Conversion of expression (prefix to infix, infix to postfix). Queue: Introduction, Types of queues (Circular Queue, Dequeue), Queue Implementation using arrays, Operations on Queue (Traversing, Insertion, deletion, and modification), Application of Queue (priority queue).	**10 Hrs**

III	**Module 3: Searching, Sorting** Searching & Sorting: Need for Searching and sorting, Linear search, binary search, bubble sort, selection sort, insertion sort.	**10 Hrs**

Text Books:

1. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill

Reference Books:

1. Data Structures using C, by Seema Threja, 2nd Edition, Oxford Press.

E-contents:

1. Fundamentals of Data Structures in C, by Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed
2. Design & Analysis of computer Algorithms by Alfred Aho, John Hopcroft and Jeffery Ullman (Link) https://doc.lagout.org/science/0_Computer Science/2_Algorithms/The Design and Analysis of Computer Algorithms %5BAho%2C Hopcroft %26 Ullman 1974-01-11%5D.pdf
3. Introduction to Algorithms by Thomas Corman et.al (Link) http://labs.xjtudlc.com/labs/wldmt/reading list/books/Algorithms and optimization/Introduction to Algorithms.pdf

DSC-8: Lab Course -IV (Based on DCS-7)

Total Credits: 02
Maximum Marks : 50

Total Contact Hours: 60 Hrs

Lab Session No.	Lab Title/Topic
1	Write an algorithm, create a flowchart, and implement C code to input elements into a 1-D array and display the array in reverse order.
2	Draft an algorithm, draw a flowchart, and write C code to read two arrays from the user, merge them, and display the merged array.
3	Develop an algorithm, design a flowchart, and implement C code to insert an element into an existing array.
4	Outline an algorithm, create a flowchart, and write C code to delete an element from an array.
5	Design an algorithm, illustrate a flowchart, and code in C to implement the linear search technique.
6	Develop an algorithm, draw a flowchart, and implement in C code to perform binary search.
7	Draft an algorithm, create a flowchart, and write C code to perform bubble sort on a list.
8	Develop an algorithm, design a flowchart, and implement C code to perform selection sort on a list.
9	Outline an algorithm, illustrate a flowchart, and code in C to perform insertion sort on a list.
10	Write an algorithm, create a flowchart, and implement C code to create a stack with Push, Pop, Display, and Exit operations.
11	Draft an algorithm, draw a flowchart, and write C code to convert an infix expression to postfix.
12	Develop an algorithm, design a flowchart, and implement C code to convert an infix expression to prefix.
13	Outline an algorithm, create a flowchart, and write C code to evaluate a postfix expression.
14	Design an algorithm, illustrate a flowchart, and code in C to implement a queue with Insert, Delete, Display, and Exit operations.
15	Develop an algorithm, draw a flowchart, and implement C code to create a circular queue.

DSC-9: C- Programming-II

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Understand array and function concepts, including initialization, accessing elements, and memory representation, alongside mastering parameter passing mechanisms like Call by Value and Call by Reference.
ii) Proficiency in utilizing structures, unions, and pointers, including dynamic memory allocation, pointer arithmetic, and the distinction between structures and unions.
iii) Master storage classes, preprocessors, and string handling functions, including scope and lifetime of variables, preprocessor directives, and string manipulation.
iv) Apply learned concepts to real-world programming scenarios, effectively utilizing arrays, functions, structures, unions, pointers, and string handling functions in C programming tasks.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Demonstrate proficiency in designing and implementing C programs utilizing arrays and functions, effectively initializing, accessing elements, and managing memory allocation.
ii) Apply advanced concepts of structures, unions, and pointers in programming tasks, including nested structures, dynamic memory allocation, and pointer arithmetic, to organize and manipulate data efficiently.
iii) Utilize storage classes, preprocessors, and string handling functions proficiently, demonstrating a clear understanding of variable scope, preprocessor directives, and effective string manipulation techniques.
iv) Develop practical problem-solving skills, applying learned concepts to real-world programming scenarios, and effectively employing arrays, functions, structures, unions, pointers, and string handling functions to create robust and efficient C programs.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module I: Arrays & Functions:** Introduction to Arrays, Declaration and initialization Accessing array elements, Memory representation of array. One & two dimension arrays, multidimensional arrays, character array, Introduction to string. Introduction to functions, Types of functions, defining functions, Arguments, Function prototype, actual parameters and formal parameters, calling function, Returning function results. Parameter Passing Mechanism: Call by Value & Call by Reference, Recursion.	**10 Hrs**
II	**Module II: Structure, Union & Pointers:** Structure: Introduction, Declaration and initializing structure, Accessing structure members, Nested structures, Arrays of structure, typedef statement and Enumerated data types. **Unions:** Declaration, Difference between structure and union. **Pointers**: Introduction, The Address (&) and Indirection (*) Operators, Declaration and initialization of pointers. Pointer expression and pointer	**10 Hrs**

	arithmetic, Pointer to pointer. Dynamic Memory Allocation in C using malloc(), calloc(), free() and realloc()	
III	**Module III: Storage classes, Preprocessors & String handling Functions: Storage classes**, Scope, visibility and lifetime of variable, block and file scope, auto, extern, static and register storage classes. **String handling functions**: strcpy(), strcmp(), strcat(), strlen(), strupr(), strlwr(), gets(), puts(), **Preprocessor Directives**: File inclusion and conditional compiler directives, Macro substitution, #define, #if, #ifdef, #else, #elif, #endif	**10 Hrs**

Text Books:

1. Kanetkar, Y.P. "Let us C", bpb publication.
2. Balaburuswamy, E. "Programming in C", Tata Macgraw Hill.
3. Goterfried. "Programming in C", Shaums' Series.

Reference Books:

1. Kooper, Moolish. "Spirit of 'C'".

E-contents:

1. Websites like GeeksforGeeks (https://www.geeksforgeeks.org/) and TutorialsPoint (https://www.tutorialspoint.com/) offer comprehensive tutorials on flowcharts, algorithms, and data structures.
2. YouTube channels such as CS Dojo (https://www.youtube.com/user/CSDojo) and freeCodeCamp.org (https://www.youtube.com/channel/UC8butISFwT-Wl7EV0hUK0BQ) provide video tutorials on these topics.
3. **C Programming**:

- For learning C programming, you can check out free online courses on platforms like Coursera (https://www.coursera.org/), edX (https://www.edx.org/), and Udemy (https://www.udemy.com/).
- There are also numerous tutorials and resources available on websites like GeeksforGeeks, TutorialsPoint, and W3Schools (https://www.w3schools.com/).
- YouTube channels like Programming with Mosh (https://www.youtube.com/user/programmingwithmosh) and TheNewBoston (https://www.youtube.com/user/thenewboston) offer video tutorials on C programming.

DSC-10: Lab Course -V (Based on DCS-09)

Total Credits: 02
Maximum Marks : 50 Total Contact Hours: 60 Hrs

Lab Session No.	Lab Title/Topic
1	Begin by drafting the algorithm for exchanging two numbers. Then, create a flowchart to illustrate the logic. Finally, implement the program using functions.
2	Outline the algorithm to find the factorial of a given number. Next, create a flowchart to visualize the process. Finally, code the program using functions.
3	Design the algorithm for creating a structure representing a student. Draw a flowchart depicting the process. Write the C program to implement it.
4	Draft the algorithm for using an array of structures. Create a flowchart to represent the logic. Implement the program in C.
5	Develop the algorithm for creating a union representing an employee. Illustrate the process with a flowchart. Write the code for the program.
6	Design the algorithm to calculate the sizes of a structure and a union. Draw a flowchart to visualize the process. Implement the program in C.
7	Draft the algorithm for using double pointers. Create a flowchart to represent the logic. Write the C program.
8	Outline the algorithm for exchanging two numbers using pointers. Draw a flowchart depicting the process. Implement the program in C.
9	Design the algorithm for demonstrating the auto and static storage classes. Draw a flowchart to illustrate the process. Write the C program.
10	Outline the algorithm for demonstrating the extern and register storage classes. Create a flowchart to visualize the steps. Code the program in C.
11	Draft the algorithm for finding the area of a circle. Create a flowchart to represent the logic. Implement the program in C.
12	Design the algorithm for demonstrating the usage of preprocessor directives like #ifdef, #if, and #elif. Draw a flowchart to visualize the steps. Code the program in C.
13	Outline the algorithm for calculating the length of a string and comparing two strings. Create a flowchart to represent the process. Implement the program in C.
14	Develop the algorithm for string copy and concatenation. Create a flowchart to visualize the logic. Write the C program.
15	Design the algorithm for finding the factorial of a given number using recursion. Draw a flowchart to represent the process. Implement the program in C.
16	Outline the algorithm for demonstrating the usage of the enum data type. Create a flowchart to illustrate the logic. Code the program in C.

DSC-11: Operating System-II

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Learn the principles underlying deadlock formation, detection, and resolution to ensure system reliability and performance stability.
ii) Master various memory allocation strategies, such as paging and segmentation, to optimize resource allocation and enhance system efficiency.
iii) Explore the logical and physical address space concepts and their implications on memory management to improve memory utilization.
iv) Understand the significance of virtual memory in modern computing systems and its role in mitigating memory constraints and enhancing multitasking capabilities.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Demonstrate proficiency in identifying, analyzing, and resolving deadlock situations in operating systems.
ii) Apply various memory allocation strategies effectively to optimize resource utilization and enhance system performance.
iii) Evaluate and implement appropriate disk management techniques to efficiently manage file systems and disk I/O operations.
iv) Design and implement device management solutions, including I/O system components and device interfaces, to facilitate seamless data transfer and device interaction within operating systems.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module I: Deadlock & Memory Management:** Deadlocks: Concept of Deadlock, Deadlock Modeling, Methods for Handling Deadlock., Address Binding, Logical Vs. Physical address space, Memory Allocation Strategies- Fixed and Variable Partitions, Paging, Segmentation, Virtual Memory.	**10 Hrs**
II	**Module II: : Disk Management:** Concept of File, File Operation, Directory Structure, File Allocation Methods- Contiguous and Non-Contiguous allocation method, **Secondary Storage Structure:** Disk fundamental, Disk Scheduling – FCFS Scheduling, SSTF Scheduling, SCAN Scheduling, Disk management.	**10 Hrs**
III	**Module III: Device Management:** Introduction: Dedicated devices, Shared devices and Virtual devices, Pipes, Buffer, I/O System Components : I/O Devices, I/O Hardware, Interrupts, Application I/O Interface.	**10 Hrs**

Text Books:

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. "Operating System Concepts." Wiley, 2021

Reference Books:

1. Tanenbaum, Andrew S., and Herbert Bos. "Modern Operating Systems." Pearson, 2020.
2. Galvin, Peter Baer, et al. "Operating System Concepts Essentials." Wiley, 2022.
3. Stallings, William. "Operating Systems: Internals and Design Principles." Pearson, 2018.
4. Singh, A. K., and Shweta Singh. "Operating System." Pearson, 2019.

E-contents:

1. **Operating System Concepts Online Exercises**: A collection of interactive exercises and quizzes based on the "Operating System Concepts" textbook. Available at https://www.os-book.com/OS10/simulation/.
2. **Operating Systems: Three Easy Pieces**: An online book by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, covering fundamental concepts in operating systems. Available for free at http://pages.cs.wisc.edu/~remzi/OSTEP/.
3. **Operating System Lecture Notes**: Lecture notes and slides on operating system concepts from various universities, available through sites like MIT OpenCourseWare, Stanford Online, and Coursera.
4. **GeeksforGeeks - Operating System**: GeeksforGeeks offers a comprehensive collection of articles, tutorials, and coding exercises on operating systems concepts. Available at https://www.geeksforgeeks.org/operating-systems/.
5. **OSDev Wiki**: A wiki dedicated to operating system development, providing tutorials, articles, and resources for those interested in creating their own operating systems. Available at https://wiki.osdev.org/Main_Page..

DSC-12: Lab Course -VI (Based on DCS-11)

Total Credits: 02
Maximum Marks : 50 Total Contact Hours: 60 Hrs

Lab Session No.	Lab Title/Topic
1	Implement memory management algorithms (first-fit, best-fit, worst-fit) in C and illustrate their operation using flowcharts.
2	Develop a program in C to implement file allocation using a linked list data structure, accompanied by the corresponding algorithm and flowchart.
3	Create a program in C to simulate and visualize the FIFO page replacement algorithm, including the algorithmic steps and flowchart representation.
4	Design a program in C to implement the LRU page replacement algorithm, providing the algorithmic details and a flowchart for visualization.
5	Implement the optimal page replacement algorithm in C with accompanying algorithmic description and flowchart representation.
6	Develop a program in C to simulate the SSTF disk scheduling algorithm, complete with algorithmic explanation and a flowchart demonstrating its operation.
7	Implement a user password setting feature at the operating system level in C, outlining the algorithmic steps and providing a flowchart for clarity.
8	Demonstrate the installation process of two peripheral devices in C, illustrating the steps with an algorithm and flowchart.
9	Develop a memory management program in C using the buddy system allocation technique, accompanied by a detailed algorithm and flowchart for understanding.
10	Create a program in C to simulate disk fragmentation and analyze its impact on disk performance, with an algorithmic explanation and flowchart representation.
11	Implement a directory structure program in C, incorporating file operations (create, read, write, delete) along with the corresponding algorithm and flowchart.
12	Develop a program in C to implement file allocation using the indexed allocation method, providing the algorithmic steps and a flowchart for visualization.
13	Create a program in C to simulate disk scheduling algorithms with a graphical user interface for visualization, accompanied by detailed algorithms and flowcharts.
14	Implement a simple device driver for a virtual device in C, providing the algorithmic steps for device management and a flowchart for clarity.
15	Develop a basic interrupt handler routine for handling hardware interrupts in C, accompanied by an algorithmic description and a flowchart demonstrating the interrupt handling process.

VSC-1 A: PC-Maintenance-1

Total Credits : 01 | Total Contact Hours : 15 Hrs

Maximum Marks : 50

Learning Objectives of the Course:

i) To get knowledge of working of a PC
ii) To get knowledge of basics of PC Maintenance and safety measures.
iii) To get knowledge of Motherboard basics, BIOS, Different Cards.
iv) To get knowledge of PC Assembly

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Perform Basic Trouble shooting.
ii) Installation of Operating System, and device drivers.
iii) Basic maintenance like Anti virus installation and updating and system scanning
iv) Upgrading a PC
v) Able to assemble a Desktop PC

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	The Four Main Functions of Computing. PC Hardware Components. PC Workspaces and Tools, Environmental and Safety Concerns. SMPS and its Types	5 Hrs
II	Motherboards and Central Processing Units, Types of Motherboards, Motherboard Form Factors, Mother Board Components : Central Processing Unit (CPU) and Processor Socket or Slot, Motherboard Buses, Chipsets, Expansion Slots, Memory Slots, Connectors	5 Hrs
III	Types of Peripheral Devices, Device Drivers, Types of RAMs (DDR2, DDR4,..), Types of Hard Drives (HDD, SSD,..) Graphics Cards: Introduction, working and types. Operating System Installation. Antivirus types.	5 Hrs

Reference Books:

1. Wiley Pathways PC Hardware Essentials Project Manual by Groth, David ; Gilster, Ron, Liberty Lake, Washington ; Polo, Russel
2. Troubleshooting, Maintaining & Repairing PCs by Stephen J. Bigelow, Tata McGraw-Hill. The Complete PC Upgrade and Maintenance Guide by Mark Minasi, BPB Publication

VSC-2- Practical Based on VSC-1 (Computer Hardware)

Total Credits : 01 Total Contact Hours : 30 Hrs

Maximum Marks : 50

List of practical to be conducted in Laboratories :

1) Practical to perform different Preventive Maintenance done for a Desktop PC
2) Practical to understand working of SMPS and trouble shooting of SMPS
3) Motherboard fitting in Cabinet, Fitting processor, RAM and other cards.
4) Fitting SMPS Disk Drives, making its connections on motherboard.
5) Entering into BIOS and exploring its various options. Setting Boot Priorities, enabling network booting.
6) Installation of Operating system Windows/Linux
7) Connecting various Peripherals to Computer and installing their drivers.
8) Installation of Antivirus, updating it, making different settings
9) Network settings: putting PC on DHCP, setting static IP, configuring internet gateway for multiple internet connections, , connection to Wi-Fi network.
10) Creating users and giving rights to it. Sharing of resources like printer, folders and giving access rights.
11) Performing PC maintenance activity like Disk cleanup, disk defragmentation, disk configuration etc.
12) Connecting PC to Projector and making its settings.

VSC-1B: Office Tools -I

Total Credits: 01 Total Contact Hours: 15 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) Understand office automation fundamentals, including its definition, history, and current trends.
ii) Master word processing for creating, formatting, and managing documents efficiently.
iii) Excel in spreadsheet software for data analysis, visualization, and database management.
iv) Learn automation techniques to streamline tasks and boost productivity in office workflows.

Course Outcomes (COs) :

After completion of the course, students will be able to -

i) Gain a comprehensive understanding of office automation principles and its practical applications in modern workplaces.
ii) Develop proficiency in using word processing tools for creating and managing various types of documents effectively.
iii) Acquire advanced skills in spreadsheet software for data analysis, visualization, and database management tasks.
iv) Demonstrate the ability to automate routine office tasks and processes to enhance productivity and efficiency in professional settings.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	**Module 1: Introduction to Office Automation** Definition and scope of office automation Benefits of office automation Overview of office automation tools and applications History of office automation and Trends in office automation	**5 Hrs**
II	**Module 2: Word Processing and Document Management** Creating and formatting basic documents Advanced formatting techniques (e.g. styles, templates, themes) Working with tables and columns Managing document content and structure Reviewing and revising documents	**5 Hrs**
III	**Module 3: Spreadsheet and Database Management** Creating and managing basic spreadsheets Advanced formatting techniques (e.g. conditional formatting, data validation) Data analysis and visualization (e.g. charts, pivot tables) Database management and design (e.g. creating tables, relationships, queries) Importing and exporting data, Automating tasks (e.g. macros, scripts)	**5 Hrs**

Text Books:

1. "Office Automation: Principles and Practice" by Dr. R. K. Singla and Dr. N. P. Singh.
2. "Office Automation and Collaboration" by Prakash Rao

Reference Books:

1. Office 2019 All-in-One For Dummies" by Peter Weverka (Wiley, 2018)
2. "Microsoft Office 2019 Inside Out" by Joe Habraken (Microsoft Press, 2019)
3. "Office 365 & Exchange Online: Essentials for Administration" by William Stanek (CreateSpace Independent Publishing Platform, 2017)
4. "Office 365 for Dummies" by Rosemarie Withee, Ken Withee, and Jennifer Reed (Wiley,

2019)

5. "The Ultimate Guide to Microsoft Office 365" by Sherri McLeish (Independently Published, 2021)

E-contents:

1. Office Automation - Overview - https://www.tutorialspoint.com/office_automation/office_automation_overview.htm
2. History and Development of Office Automation - https://www.guru99.com/office-automation.html
3. The Advantages of Office Automation - https://smallbusiness.chron.com/advantages-office-automation-3077.html
4. Microsoft Word Basics - https://edu.gcfglobal.org/en/wordbasics/
5. Advanced Microsoft Word - https://edu.gcfglobal.org/en/advanced-word/
6. Microsoft Excel Basics - https://edu.gcfglobal.org/en/excelbasics/
7. Advanced Microsoft Excel - https://edu.gcfglobal.org/en/advanced-excel/
8. Database Management Basics - https://www.guru99.com/database-management-system.html
9. Google Meet Basics - https://edu.gcfglobal.org/en/google-meet/
10. Microsoft Teams Basics - https://edu.gcfglobal.org/en/microsoft-teams/
11. Basic Internet Security - https://www.gcflearnfree.org/internetsafety/basic-internet-security/

VSC-2: Lab Course -I (Based on VSC-1B)

Total Credits: 01
Maximum Marks : 50 Total Contact Hours: 30 Hrs

Lab Session No.	Lab Title/Topic
1	**Lab 1: Introduction to Office Automation Tools** Objective: Familiarize students with various office automation tools and their applications. Task: Explore different office automation software suites such as Microsoft Office, Google Workspace, and other specialized tools.
2	**Lab 2: History and Trends in Office Automation** Objective: Understand the historical development and current trends in office automation. Task: Research and present key milestones in the evolution of office automation technology and identify emerging trends.
3	**Lab 3: Benefits of Office Automation** Objective: Identify and analyze the benefits of implementing office automation in business environments. Task: Conduct case studies or interviews with professionals to explore the impact of office automation on productivity, efficiency, and cost savings
4	**Lab 4: Creating Basic Documents** Objective: Learn fundamental skills for creating and formatting basic documents. Task: Create simple documents using word processing software, focusing on text formatting, alignment, and basic page layout.
5	**Lab 5: Advanced Document Formatting Techniques** Objective: Explore advanced formatting features to enhance document design and efficiency. Task: Experiment with styles, templates, and themes to create professional-looking documents with consistent formatting.
6	**Lab 6: Working with Tables and Columns** Objective: Develop proficiency in creating and formatting tables and columns in documents. Task: Create tables for organizing data and information, and explore options for customizing table layouts and styles.
7	**Lab 7: Document Content and Structure Management** Objective: Learn strategies for managing document content and structure effectively. Task: Practice organizing and structuring document content using headings, sections, and navigation features.
8	**Lab 8: Reviewing and Revising Documents** Objective: Understand the importance of document review and revision processes. Task: Collaborate with peers to review and provide feedback on each other's documents, focusing on clarity, accuracy, and coherence.
9	**Lab 9: Document Sharing and Collaboration** Objective: Explore tools and techniques for sharing and collaborating on documents remotely. Task: Use cloud-based collaboration platforms such as Google Docs or Microsoft Teams to co-author documents and track changes.

10	**Lab 10: Automating Document Creation** Objective: Learn how to automate repetitive tasks and streamline document creation processes. Task: Explore features such as mail merge and macros to automate document generation tasks and improve workflow efficiency.
11	**Lab 11: Basic Spreadsheet Creation and Management** Objective: Develop basic skills for creating and managing spreadsheets. Task: Create simple spreadsheets for organizing data, performing basic calculations, and creating charts.
12	**Lab 12: Advanced Spreadsheet Formatting and Analysis** Objective: Explore advanced formatting techniques and data analysis tools in spreadsheets. Task: Apply conditional formatting, data validation, and pivot tables to analyze and visualize data effectively.
13	**Lab 13: Database Management Fundamentals** Objective: Understand the fundamentals of database management, including data organization and querying. Task: Design and create a basic database using database management software, and practice querying data.
14	**Lab 14: Importing and Exporting Data** Objective: Learn how to import data from external sources and export data for use in other applications. Task: Import data from spreadsheets or text files into a database, and export query results to different file formats.
15	**Lab 15: Task Automation with Macros and Scripts** Objective: Explore methods for automating tasks and processes using macros and scripts. Task: Create and customize macros or scripts to automate repetitive tasks in office automation software, and evaluate their effectiveness.

This course will be available for the students from other faculty

GE/OE-2: E-Commerce

Total Credits: 02 Total Contact Hours: 30 Hrs
Maximum Marks : 50

Learning Objectives of the Course:

i) After the successful completion of the course the student must be aware of Techniques in Application of e-commerce.
ii) This course is designed to provide basic knowledge about Electronic Commerce.

Course Outcomes (COs) :

i) Student will get knowledge of different aspects of E-Commerce
ii) Able to identify different components of E-Commerce
iii) Will get knowledge of different security technologies used in E-Commerce.
iv) Understand the risk in the e-commerce business.

Module No.	Topics / actual contents of the syllabus	Contact Hours
I	Introduction, IT and business, E-commerce: Concepts, Benefits of E-Commerce; Types of E-Commerce, Advantages and Disadvantage of E-commerce, Electronic Communication, PCs and Networking, E-mail, Internet and intranets.	**10 Hrs**
II	EDI to E-commerce, EDI, UN/EDIFACT. Concerns for E-commerce Growth, Internet bandwidth, Technical issues, Security issues, Legal issues. Roadmap of e-commerce in India.	**10 Hrs**
III	Security Technologies: Encryption, Symmetric key Encryption, Public key encryption, Public key encryption using digital Signatures. Hashing techniques, Certification and key Distribution.	**10 Hrs**

Text Books:

1. E-Commerce: The Cutting Edge of Business, Kamlesh K. Bajaj & Debjani Nag, Tata McGraw Hill.
2. E- Commerce Strategy , Technologies and Applications, David Whiteley, McGraw Hill Edition