CURRICULUM FRAMEWORK FOR THREE YEAR UG DEGREE / FOUR YEAR UG DEGREE (HONOURS) / FOUR YEAR UG DEGREE (HONOURS WITH RESEARCH)

Bachelor in Computer Application

Department of Computer Science
North Lakhimpur University

(As per NEP 2020 Guidelines and AICTE Model Curriculum for BCA, 2024)

(As approved by Board of Studies in Computer Science held on 18th June 2026 and passed by Academic Council held on .....)

Contents

1.0 Introduction: 1

1.1 Nature and extent of the B.C.A. program 1
1.2 Aims of bachelor's degree Programme in Computer Applications (BCA) 2
1.3 Graduate Attributes of Bachelor of Computer Applications (BCA) program 2
1.4 Qualification Descriptors for Bachelor of Computer Applications (BCA) program 3
1.5 Programme Specific Outcome 4
1.6 Teaching-Learning and Assessment Methodology 5

2.0 Certification Criteria 5

2.1 Exit and Re-entry 5
2.2 Internship, Community Engagement and Capstone Components 6

3.0 General Course Structure and Credit Distribution 6

3.1 Definition of Credit 6
3.2 Course Label and Definition 6
3.3 Course Name, Duration and Eligibility 6
3.4 Semester-wise Credit Distribution 8
3.5 Category-wise Credit Distribution 8

4.0 Semester-wise Structure and Curriculum 9

SEMESTER I 9
SEMESTER II 9
SEMESTER III 10
SEMESTER IV 10
SEMESTER V 11
SEMESTER VI 11
SEMESTER VII - BCA (Honours), Specialization: AI & ML 12
SEMESTER VII - BCA (Honours), Specialization: Data Science 12
SEMESTER VIII - BCA (Honours) 12
SEMESTER VII - BCA (Honours with Research) 13
SEMESTER VIII - BCA (Honours with Research) 13

5.0 Proposed Discipline-Specific Elective Streams 14

5.1 Indicative Professional Electives 14

CC-3-BCA-101: Mathematics Foundation to Computer Science - I 15

CC-5-BCA-102: Computer Architecture 18

CC-2-BCA-103: Entrepreneurship and Startup Ecosystem 22

SEC–5-BCA-101: Problem Solving Techniques 25

IKS–2-BCA-101: Indian Mathematics and Astronomy OLD 29

MDE-2-BCA-101: Indian Knowledge System in Computer Science (NEW) 29

CC-3-BCA-201: Mathematics Foundations to Computer Science - II 33

CC-5-BCA-202: Data Structure 36

CC-3-BCA-203: Probability and Statistics 39

SEC-5-BCA-201: Object Oriented Programming using Java 42

SEC-2-BCA-202: Web Technologies 46

CC-4-BCA-301: Operating Systems 49

CC-5-BCA-302: Database Management System 52

CC-3-BCA-303: Software Engineering 59

SEC-4-BCA-301: Python Programming 62

VAC-2-BCA-301: Indian Constitution 66

CC-5-BCA-401: Computer Networks 70

CC-3-BCA-402: Design and Analysis of Algorithms 74

CC-5-BCA-403: Artificial Intelligence 77

SEC-2-BCA-401: : Design Thinking and Innovation 81

SEC-2-BCA-501: Quantitative Techniques 85

SEC-4-BCA-502: Internship/Capstone Project 88

SEC-4-BCA-601: Major Project - Phase I: Initiation and Design 91

CC-4-BCA-601: Generative AI: Generative AI 92

AEC-1-BCA-601 Soft Skills: Soft Skills 95

SEC-4-BCA-601: Major Project - Phase II: Implementation and Evaluation 98

DSE-3-BCA-301: Feature Engineering 101

DSE-3-BCA-401: Introduction to Machine Learning 104

DSE-5-BCA-501: Neural Network 107

DSE-5-BCA-502: Digital Image Processing 110

DSE-5-BCA-503: Natural Language Processing 113

DSE-5-BCA-601: Deep Learning for Computer Vision 116

DSE-5-BCA-602: Predictive Analysis 119

DSE-3-BCA-301-D: Basics of Data Analytics using Spreadsheet 122

DSE-3-BCA-401-D: Data Visualization 125

DSE-5-BCA-501-D: Introduction to Data Science 128

DSE-5-BCA-502-D: Time Series Analysis 131

DSE-5-BCA-503-D: Machine Learning 134

DSE-5-BCA-601-D: Big Data Analytics 137

DSE-5-BCA-602-D: Exploratory Data Analysis 140

DSE-3-BCA-301-F: Web Programming - I 143

DSE-3-BCA-401-F: Web Programming - II 146

1.0 Introduction:

The Bachelor of Computer Applications (BCA) program is a dynamic and industry-relevant undergraduate degree designed to provide students with a strong foundation in computer science, software development, and information technology. This program equips students with essential technical skills, programming expertise, and computational thinking, preparing them for diverse careers in the ever-evolving tech industry.

Blending theoretical concepts with hands-on practical applications, the BCA curriculum covers software engineering, database management, web technologies, artificial intelligence, and cyber security. Students gain valuable experience through coding projects, and internships, enhancing their problem-solving abilities and technological proficiency.

Aligned with the New Education Policy (NEP) 2020, the program follows a multidisciplinary approach, integrating core computing principles with data-driven decision-making, digital innovation, and entrepreneurial skills. Emphasizing ICT applications, leadership, ethics, and sustainability, the BCA degree fosters critical thinking and adaptability in an era of rapid technological advancements.

With the increasing demand for IT professionals, BCA graduates can pursue careers in software development, data analytics, IT consulting, cybersecurity, and digital transformation. They can also opt for higher studies in computer science, artificial intelligence, or related IT fields to specialize in cutting-edge fields.

Ideal for students with a keen interest in technology, problem-solving, and software development, the BCA program serves as a gateway to a successful and future-ready career in the digital world.

1.1 Nature and extent of the B.C.A. program

The National Curriculum Framework for Undergraduate Studies in Computer Science is an outcome-based framework designed to establish academic standards and expected learning outcomes for graduates. This framework ensures that students acquiring a qualification in Computer Science develop the necessary technical knowledge, problem-solving skills, and computational competencies to meet both academic and professional demands.

The curriculum is structured around key components such as Graduate Attributes, Qualification Descriptors, Programme Learning Outcomes, and Course-Level Learning Outcomes. These elements serve as guiding principles for curriculum planning at the undergraduate level.

The overall objectives of the learning outcomes-based curriculum framework are to:

- Define and articulate graduate attributes, qualification descriptors, and learning outcomes at both program and course levels, ensuring clarity in academic expectations.
- Provide students, parents, employers, and stakeholders with a clear understanding of the technical skills, problem-solving abilities, ethical considerations, and professional values that a graduate is expected to demonstrate upon program completion.
- Maintain national academic standards while ensuring international comparability, enhancing the global competitiveness of graduates and supporting student mobility across institutions and countries.
- Assist higher education institutions in designing effective teaching-learning strategies, student assessment methods, and periodic curriculum reviews to maintain academic excellence and relevance in the rapidly evolving IT sector.

The BCA program integrates theoretical foundations, hands-on practical skills, and problem-solving expertise, equipping graduates for higher studies, research, and diverse career opportunities in software development, artificial intelligence, cloud computing, cybersecurity, and emerging digital technologies. By aligning with global standards, the program ensures that graduates remain technologically competent, adaptable, and industry-ready in an ever-evolving digital landscape.

1.2 Aims of bachelor's degree Programme in Computer Applications (BCA)

The Bachelor of Computer Applications (BCA) program aims to provide students with a strong foundation in computer science, software development, and information technology, equipping them with the necessary skills for academic and professional growth. The program is designed to meet national and international standards, ensuring graduates are well-prepared for diverse career opportunities and higher studies in emerging technology domains.

The key aims of the program are to:

- Develop a comprehensive understanding of computer science principles, including programming, software development, database management, networking, cyber-security, and artificial intelligence.
- Foster critical thinking, problem-solving abilities, and analytical reasoning to enable students to address real-world technological challenges using computational tools and methodologies.
- Provide hands-on experience with modern programming languages, development frameworks, and IT infrastructure, enhancing students' ability to design, develop, and implement innovative software solutions.
- Promote the application of computing and IT solutions in various domains such as business, finance, healthcare, e-commerce, education, and smart technologies.
- Equip students with research skills and the ability to analyze, design, and develop software applications, incorporating project-based learning and industry internships.
- Align with the National Curriculum Framework and New Education Policy (NEP) 2020, ensuring a multidisciplinary, skill-based, and industry-oriented approach to learning.
- Instill ethical values, professional responsibility, and a commitment to lifelong learning, fostering a responsible approach to cybersecurity, data privacy, and technology ethics.
- Enhance communication, teamwork, leadership, and ICT skills, preparing students for diverse roles in software engineering, IT consulting, system administration, data analytics, cloud computing, and entrepreneurship.
- Ensure global competitiveness by maintaining academic excellence, facilitating international mobility, and enabling graduates to pursue advanced studies in computer science, artificial intelligence, machine learning, and emerging technologies.

By achieving these aims, the BCA program ensures that graduates are well-equipped to contribute to industry, research, and entrepreneurship, fostering innovation, digital transformation, and technology-driven solutions in the modern world.

1.3 Graduate Attributes of Bachelor of Computer Applications (BCA) program

The Bachelor of Computer Applications (BCA) program aims to develop well-rounded graduates with a strong foundation in computer science, software development, and emerging technologies. The graduate attributes define the key competencies, skills, and values that students are expected to demonstrate upon completion of the program.

- Disciplinary Knowledge: Graduates will have a comprehensive understanding of computer science principles, programming, algorithms, data structures, software engineering, and IT infrastructure, along with their real-world applications.
- Analytical and Problem-Solving Skills: Graduates will be able to analyze complex problems, develop algorithms, and implement efficient solutions using computational and logical reasoning in various domains.
- Research and Innovation: Students will develop a research-oriented mindset and the ability to explore, design, and innovate software solutions, AI-driven applications, and emerging technologies to address industry challenges.
- Computational and ICT Proficiency: Graduates will gain proficiency in programming languages, databases, cloud computing, artificial intelligence, cyber-security, and modern development frameworks, enabling them to build scalable and efficient IT solutions.
- Critical Thinking and Logical Reasoning: The program nurtures logical thinking, debugging skills, and system optimization techniques, ensuring students can critically evaluate technological problems and propose effective solutions.
- Communication and Presentation Skills: Graduates will develop the ability to effectively communicate technical concepts, write project reports, and present ideas to technical and non-technical audiences through various digital platforms.
- Ethical Awareness and Professional Responsibility: Students will be instilled with ethical principles in data security, privacy laws, intellectual property rights, and responsible AI, ensuring professional integrity and compliance with global IT standards.
- Leadership, Teamwork, and Interpersonal Skills: Graduates will demonstrate teamwork, collaboration, and leadership abilities, essential for working in multidisciplinary environments and managing software projects.
- Lifelong Learning and Adaptability: With rapidly evolving technological landscapes, students will develop a lifelong learning mindset, staying updated with the latest advancements in AI, cyber-security, blockchain, IoT, and cloud computing.
- Global Competence and Societal Contribution: Graduates will be equipped to apply computing knowledge for societal development, addressing real-world challenges in healthcare, smart cities, and sustainability, aligning with Sustainable Development Goals (SDGs).

By acquiring these attributes, graduates of the BCA program will be well-prepared for higher studies, research, industry roles, and entrepreneurial ventures in the ever-expanding field of technology and digital transformation.

1.4 Qualification Descriptors for **Bachelor of Computer Applications (BCA) program**

The Qualification Descriptors for the BCA outline the key competencies, skills, and knowledge that graduates are expected to demonstrate upon completion of the program. These descriptors ensure alignment with national and international academic standards, fostering academic excellence, employability, and research potential in the evolving field of technology.

A. **Comprehensive Knowledge and Understanding**: Graduates will have a broad and in-depth understanding of computer science principles, programming paradigms, software engineering, database management, cloud computing, and artificial intelligence, along with their applications in various industries.

B. **Analytical and Problem-Solving Skills**: Graduates will develop the ability to analyze complex computational problems, design algorithms, optimize systems, and develop efficient software solutions for real-world challenges in business, healthcare, finance, and smart technologies.
C. **Research and Inquiry-Based Learning**: The program prepares students to conduct independent research, explore emerging technologies, innovate in AI and cybersecurity, and apply computational methodologies to advance the field of computer science.
D. **Technical and Computational Proficiency**: Graduates will gain expertise in modern programming languages (Python, Java, C++, etc.), full-stack development, machine learning, cloud platforms, blockchain, and DevOps tools, enabling them to build and manage scalable IT solutions.
E. **Communication and Technical Documentation**: Students will be able to effectively communicate technical concepts, document software solutions, and present findings through reports, presentations, and visual representations, catering to both technical and non-technical audiences.
F. **Ethical and Social Responsibility**: Graduates will understand the ethical implications of technology, including cybersecurity, data privacy laws, AI ethics, and digital governance, ensuring the responsible use of technology for societal benefit.
G. **Lifelong Learning and Adaptability**: The program fosters a lifelong learning mindset, enabling graduates to stay updated with emerging technologies, adapt to industry trends, and continuously upgrade their technical and managerial skills.
H. **Employability and Career Readiness**: Graduates will be well-prepared for diverse career opportunities in software development, IT consulting, cybersecurity, cloud computing, AI/ML, fintech, and digital transformation, or for further academic pursuits in computer science, data analytics, and AI research.
I. **Multidisciplinary and Global Perspectives**: The curriculum integrates multidisciplinary approaches and international academic standards, ensuring graduates possess a global outlook and the ability to work across industries, research institutions, and multinational organizations worldwide.

By achieving these qualification descriptors, graduates of the BCA program will be equipped with the expertise and skills required for professional success, research excellence, and technological innovation, making meaningful contributions to the digital economy and global technology landscape.

1.5 **Programme Specific Outcome**

Programme Specific outcomes relating to bachelor degree in Computer Science/Application are as follows:

1. **Computational Knowledge:** Apply knowledge of computing fundamentals, mathematics, and domain-specific knowledge to solve complex problems.
2. **Problem Analysis:** Identify, formulate, and analyze problems using structured and unstructured data, reaching substantiated conclusions using data analytics and computing tools.
3. **Design and Development of Solutions:** Design and evaluate software solutions that meet user-defined needs with appropriate consideration for public health, safety, culture, and environment.
4. **Modern Tool Usage:** Create, select, and apply modern computing tools and techniques for software development with an understanding of their limitations.

5. **Ethics:** Understand and commit to professional ethics, cyber regulations, and responsibilities in computing practice.
6. **Communication:** Communicate effectively in both verbal and written forms, including documentation, presentation, and instruction writing.
7. **Individual and Team Work:** Function effectively as an individual and as a member or leader in diverse teams and multidisciplinary settings.
8. **Project Management and Finance:** Demonstrate knowledge and understanding of management principles and apply them in software project management and entrepreneurial ventures.
9. **Life-long Learning:** Recognize the need for and engage in independent and life-long learning in the context of technological change.
10. **Societal and Environmental Concern:** Understand the impact of professional solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

These outcomes help ensure that BCA graduates are equipped with the necessary technical, problem-solving, and professional skills to excel in the fast-evolving field of computer science.

1.6 Teaching-Learning and Assessment Methodology

Teaching-learning methods shall include lectures, laboratory demonstrations, coding exercises, group discussions, peer programming, assignments, presentations, workshops, seminars, technical talks, hackathons, internships and project-based learning.

Assessment shall be aligned with programme and course learning outcomes. It may include written and oral examinations, open-book/closed-book tests, coding assignments, laboratory work, project reports, technical presentations, viva voce, internship evaluation and continuous internal assessment as per University regulations.

2.0 Certification Criteria

- UG Certificate in Computer Application: Awarded after successful completion of the first year and additional credits/requirements prescribed by the University.
- UG Diploma in Computer Application: Awarded after successful completion of the second year and additional credits/requirements prescribed by the University.
- Bachelor in Computer Application (BCA): Awarded after successful completion of the third year with 120 credits.
- BCA (Honours): Awarded after successful completion of the four-year honoursprogramme with 160 credits.
- BCA (Honours with Research): Awarded after successful completion of the four-year programme with 160 credits including prescribed research/dissertation components.

2.1 Exit and Re-entry

The programme shall follow multiple entry and exit provisions as per NEP 2020, AICTE model curriculum and University/Admitting Body rules. Students exiting after the first year may re-enter the third semester; students exiting after the second year may re-enter the fifth semester; and students exiting after the third year may re-enter the seventh semester for BCA (Honours) or BCA (Honours with Research), subject to applicable University regulations and completion of required credits.

2.2 Internship, Community Engagement and Capstone Components

1. After the first year, students may undertake skill-based/vocational work, internship/apprenticeship, or social responsibility and community engagement as prescribed by the University.
2. After the second year, students may undertake skill-based/vocational work, internship/apprenticeship, social responsibility and community engagement, or a capstone project as prescribed by the University.
3. After the fourth semester, every student shall complete the prescribed summer training/internship/capstone of eight weeks. Evaluation shall be based on report, presentation, viva voce and work quality.
4. For the fourth year, internship, dissertation and research components shall be governed by the regulations of North Lakhimpur University.

3.0 General Course Structure and Credit Distribution

3.1 Definition of Credit

Academic Component	Credit
1 hour Lecture (L) per week	1 Credit
1 hour Tutorial (T) per week	1 Credit
1 hour Practical (P) per week	0.5 Credit
2 hours Practical (P) per week	1 Credit

3.2 Course Label and Definition

Course Code	Definition
L	Lecture
T	Tutorial
P	Practical
CC	Core Courses
AEC	Ability Enhancement Courses
MDE	Multi-Disciplinary Elective Course
VAC	Value Added Courses
SEC	Skill Enhancement Courses
DSE	Discipline Specific Elective
OE	Open Elective

3.3 Course Name, Duration and Eligibility

Course Name: Bachelor in Computer Application, Bachelor in Computer Application (Honours) and Bachelor in Computer Application (Honours with Research).

Course Level/Duration/System: Undergraduate / three or four years / 6 or 8 semesters with multiple entry and exit. The following options shall be available to students joining the BCA programme:

1. One year: Under Graduate Certificate in Computer Application.
2. Two years: Under Graduate Diploma in Computer Application.
3. Three years: Bachelor in Computer Application (BCA).

4. Four years: Bachelor in Computer Application with Honours: BCA (Honours) or Bachelor in Computer Application Honours with Research: BCA (Honours with Research).

Minimum eligibility criteria for opting for the fourth year shall be as per University/Admitting Body rules. The minimum academic requirement in the AICTE model curriculum is BCA Degree for BCA (Honours) and BCA (Honours with Research).

b

3.4 Semester-wise Credit Distribution

Semester	Core Courses	Ability Enhancement Courses	Multi-Disciplinary Elective Course	Value Added Courses	Skill Enhancement Courses	Discipline Specific Elective	Total
I	10	2	2	0	5	-	19
II	12	-	-	2	7	-	21
III	12	0	0	2	4	3	21
IV	12	0	0	2	2	3	19
V	0	0	0	0	6	15	21
VI	4	1	0	0	4	10	19
BCA (Honours) - VII	5	0	3	0	4	8	20
BCA (Honours) - VIII	8	-	-	-	-	12	20
BCA (Honours with Research) - VII	12	-	-	-	-	8	20
BCA (Honours with Research) - VIII	20	-	-	-	-	-	20

3.5 Category-wise Credit Distribution

Programme	Core Courses	Ability Enhancement Courses	Multi-Disciplinary Elective Course	Value Added Courses	Skill Enhancement Courses	Discipline Specific Elective	Total
BCA	50	3	2	6	28	31	120
BCA (Honours)	55	3	5	6	40	51	160
BCA (Honours with Research)	82	3	2	6	28	39	160

Total Credits: 3-Year BCA Programme = 120; 4-Year BCA (Honours) and BCA (Honours with Research) = 160. Students may take extra credit courses from their own department or from other departments as per University norms.

4.0 Semester-wise Structure and Curriculum

SEMESTER I

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	CC101	CC-3-BCA-101	Mathematics Foundations to Computer Science - I	3	0	0	3
2	CC102	CC-5-BCA-102	Computer Architecture	3	0	4	5
3	CC103	CC-2-BCA-103	Entrepreneurship and Startup Ecosystem	1	1	0	2
4	SEC101	SEC-5-BCA-101	Problem Solving Techniques	3	0	4	5
5	AEC101	**	ENGLISH				2
6	MDE101	MDE-2-BCA-101	Indian Knowledge System^	2	0	0	2
			TOTAL				**19**

** As per the common syllabus conducted by the English Department for BA/BSc programmes.

SEMESTER II

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	CC104	CC-4-BCA-201	Mathematics Foundations to Computer Science - II	4	0	0	4
2	CC105	CC-5-BCA-202	Data Structures	3	0	4	5
3	CC106	CC-3-BCA-203	Probability and Statistics	3	0	0	3
4	SEC102	SEC-5-BCA-201	Object Oriented Programming using Java	3	0	4	5
5	SEC103	SEC-2-BCA-202	Web Technologies	1	0	2	2
6	VAC101	*	Yoga/Sports/NCC/NSS/Disaster Management	0	0	4	2
7	AEC102		Additional Course - Indian or Foreign Language other than English (1-1-0) [optional]**	1	1	0	0*
			TOTAL				**21**

* As per the common syllabus conducted by the Physical Education Department for BA/BSc programmes.

***Optional language courses may include Sanskrit, Hindi, regional languages or foreign languages as per University norms.*

SEMESTER III

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	CC201	CC-4-BCA-301	Operating Systems	3	0	2	4
2	CC202	CC-5-BCA-302	Data Base Management System	3	0	4	5
3	CC203	CC-3-BCA-303	Software Engineering	3	0	0	3
4	SEC201	SEC-2-BCA-301	Python Programming	2	0	4	4
5	DSE201*		Professional Elective - I	1	0	4	3
6	VAC201	VAC-2-BCA-301	Indian Constitution	2	0	0	2
			TOTAL				**21**

**DSE to be selected from the proposed discipline-specific elective streams such as Data Science, Artificial Intelligence and Machine Learning, and Full Stack Development.*

SEMESTER IV

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	CC204	CC-4-BCA-401	Computer Networks	3	0	2	4
2	CC205	CC-3-BCA-402	Design and Analysis of Algorithm	3	0	0	3
3	CC206	CC-5-BCA-403	Artificial Intelligence	3	0	4	5
4	SEC202	SEC-2-BCA-401	Design Thinking and Innovation	1	1	0	2
4	DSE202*		Professional Elective - II	1	0	4	3
5	VAC202**		Environmental Science and Sustainability	2	0	0	2
6	AEC201		Additional Course - Indian or Foreign Language other than English (1-1-0) [optional]	1	1	0	0*
			TOTAL				**19**

**Students must select their Discipline Specific Elective (DSE) from specialized streams—such as Data Science, Artificial Intelligence & Machine Learning, or Full Stack Development—that align with the discipline they chose in the 1st semester.*

** As per the common syllabus conducted by the other Department for BA/BSc programmes.

Note: At the end of the fourth semester, every student shall undergo summer training/internship/capstone for eight weeks in industry, research or academic institute. This component shall be evaluated during the fifth semester.

SEMESTER V

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	DSE301*		Professional Elective - III	3	0	4	5
2	DSE302*		Professional Elective - IV	3	0	4	5
3	DSE303*		Professional Elective - V	3	0	4	5
4	SEC301	SEC-2-BCA-501	Quantitative Techniques	0	2	0	2
5	SEC302	SEC-4-BCA-502	Internship/Capstone Project	0	0	8	4
6	SEC303	SEC-4-BCA-601	Major Project [Evaluation in Sixth Semester]	-	-	2	0
			TOTAL				**21**

**L-T-P distribution for electives depends on student selection. DSE streams (e.g., Data Science, AI & ML) must align with the core discipline chosen in Semesters 1 and 2.*

SEMESTER VI

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	CC301	CC-4-BCA-601	Generative AI	2	0	4	4
2	DSE304*		Professional Elective - VI	3	0	4	5
3	DSE305*		Professional Elective - VII	3	0	4	5
4	AEC301	AEC-1-BCA-601	Soft Skills	0	1	0	1
5	SEC304	SEC-4-BCA-601	Major Project [Initiated in Fifth Semester]	0	0	8	4
			TOTAL				**19**

**L-T-P distribution for electives depends on student selection. DSE streams (e.g., Data Science, AI & ML) must align with the core discipline chosen in Semesters 1 and 2.*

SEMESTER VII - BCA (Honours), Specialization: AI & ML

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	MDE401	MDE-3-BCA-701	Social Network Analysis	-	-	-	3
2	CC401	CC-5-BCA-701	Optimization of ML	3	-	4	5
3	DSE401*		Professional Elective - VIII	3	-	4	5
4	DSE402*		Professional Elective - IX	-	-	-	3
5	SEC401	SEC-8-BCA-801	Dissertation Work [Evaluation in Eighth Semester]	-	-	-	-
6	SEC402	SEC-4-BCA-701	Summer Internship II	0	0	8	4
			TOTAL				**20**

**L-T-P distribution for electives depends on student selection. DSE streams (e.g., Data Science, AI & ML) must align with the core discipline chosen in Semesters 1 and 2.*

SEMESTER VII - BCA (Honours), Specialization: Data Science

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	MDE401	MDE-3-BCA-701-D	Advanced Statistical Methods for Data Science	-	-	-	3
2	CC401	CC-5-BCA-701-D	Python for Data Science	3	-	4	5
3	DSE401*		Professional Elective - VIII	3	-	4	5
4	DSE402*		Professional Elective - IX	-	-	-	3
5	SEC401	SEC-8-BCA-801	Dissertation Work [Evaluation in Eighth Semester]	-	-	-	-
6	SEC402	SEC-4-BCA-701	Summer Internship II	0	0	8	4
			TOTAL				**20**

**L-T-P distribution for electives depends on student selection. DSE streams (e.g., Data Science, AI & ML) must align with the core discipline chosen in Semesters 1 and 2.*

SEMESTER VIII - BCA (Honours)

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	DSE403*		Professional Elective - X	3	-	4	5
2	DSE404*		Professional Elective - XI	3	-	4	5
3	DSE405*		Professional Elective - XII	2	-	-	2
4	SEC403	SEC-8-BCA-801	Dissertation Work [Started in Seventh Semester]	0	0	16	8
			TOTAL				**20**

**L-T-P distribution for electives depends on student selection. DSE streams (e.g., Data Science, AI & ML) must align with the core discipline chosen in Semesters 1 and 2.*

SEMESTER VII - BCA (Honours with Research)

S. No.	Course Label	Course Code	Course Title	L	T	P	Credit
1	CC401	CC-4-BCA-701-R	Advanced Data Analysis Tools	0	2	4	4
2	CC402	CC-4-BCA-702-R	Research Methodology	2	2	0	4
3	CC403	CC-4-BCA-703-R	Research Internship Report and Viva Voce	0	0	8	4
4	DSEXX	DSE-3-BCA-702 Or DSE-3-BCA-702-D	Professional Elective - IX	-	-	-	3
5	DSEXX	DSE-3-BCA-801 Or DSE-3-BCA-801-D	Professional Elective - X	-	-	-	5
			TOTAL				**20**

**L-T-P distribution for electives depends on student selection. DSE streams (e.g., Data Science, AI & ML) must align with the core discipline chosen in Semesters 1 and 2.*

SEMESTER VIII - BCA (Honours with Research)

S. No.	Course Label		Course Title	L	T	P	Credit
1	SEC401	SEC-20-BCA-801-R	Dissertation (For Research Track)*	-	-	-	20
			TOTAL				**20**

**The dissertation work will start from the beginning of the fourth year of the BCA (Honours with Research) programme. Students shall be assessed for project work, research internship report, viva voce and dissertation as per University norms.*

5.0 Proposed Discipline-Specific Elective Streams

The elective streams listed below are indicative. North Lakhimpur University reserves the right to add, remove, or revise these streams based on academic requirements, industry relevance, and regulatory provisions. ***Furthermore, the Department of Computer Science will finalize specific stream offerings based on faculty availability and student demand.***

- Data Science
- Artificial Intelligence and Machine Learning
- Full Stack Development

5.1 Indicative Professional Electives

Stream	Course Label	Course Code	Professional Elective
AI & ML	DSE201	DSE-3-BCA-301	Feature Engineering
	DSE202	DSE-3-BCA-401	Introduction to ML
	DSE301	DSE-5-BCA-501	Neural Network
	DSE302	DSE-5-BCA-502	Digital Image Processing
	DSE303	DSE-5-BCA-503	Natural Language Processing
	DSE304	DSE-5-BCA-601	Deep Learning for Computer Vision
	DSE305	DSE-5-BCA-602	Predictive Analysis
	DSE401	DSE-5-BCA-701	Explainable AI
	DSE402	DSE-3-BCA-702	Evolutionary Algorithm
	DSE403	DSE-5-BCA-801	Speech Recognition
	DSE404	DSE-5-BCA-802	Augmented Reality and Virtual Reality
	DSE405	DSE-2-BCA-803	Security Aspects of ML
Data Science	DSE201	DSE-3-BCA-301-D	Basics of Data Analytics using Spreadsheet
	DSE202	DSE-3-BCA-401-D	Data Visualization
	DSE301	DSE-5-BCA-501-D	Introduction to Data Science
	DSE302	DSE-5-BCA-502-D	Time Series Analysis
	DSE303	DSE-5-BCA-503-D	Machine Learning
	DSE304	DSE-5-BCA-601-D	Big Data Analytics
	DSE305	DSE-5-BCA-602-D	Exploratory Data Analysis
	DSE401	DSE-5-BCA-701-D	Business Intelligence and Analytics
	DSE402	DSE-3-BCA-702-D	Data Mining and Warehousing
	DSE403	DSE-5-BCA-801-D	Advanced Data Visualization
	DSE404	DSE-5-BCA-802-D	Cloud Computing for Data Analytics
	DSE405	DSE-2-BCA-803-D	Data Security and Privacy
Full Stack Development	DSE201	DSE-3-BCA-301-F	Web Programming - I
	DSE202	DSE-3-BCA-401-F	Web Programming - II

Semester-I

Course Code : CC-3-BCA-101
Course Title : Mathematics Foundation to Computer Science - I
Nature of the Course : CORE
Total Credit assigned : 3
Component of the Course : Lecture: 3 Practical: 0 Tutorial: 0

Course Objectives:

Objective-1. To understand and apply the concepts of sets, relations, and functions.
Objective -2. Solve counting problems and recurrence relations relevant to computer science.
Objective -3. Analyze basic properties and structures of graphs and trees.
Objective -4. Apply matrix algebra techniques to solve linear systems and related problems.

Course Content:

Unit	Content	L	P	T
I	**Set, Relation and Function:** • Set, Set Operations, Properties of Set operations, Subset, Venn Diagrams, Cartesian Products. • Relations on a Set, Properties of Relations, Representing Relations using matrices and digraphs, Types of Relations, Equivalence Relation, Equivalence relation and partition on set, Closures of Relations. • Functions, properties of functions (domain, range), composition of functions, surjective (onto), injective (one-to-one) and bijective functions, inverse of functions. • Some useful functions for Computer Science: Exponential and Logarithmic functions, Polynomial functions, Ceiling and Floor functions.	12	**0**	**0**
II	**Counting and Recurrence Relation:** • Basics of counting, Pigeonhole principle, permutation, combination, Binomial coefficients, Binomial theorem. • Recurrence relations, modelling recurrence relations with examples, like Fibonacci numbers, the tower of Hanoi problem. Solving linear recurrence relation with constant coefficients using characteristic equation roots method.	**10**	**0**	**0**
III	**Matrix Algebra:** • Types of matrices, algebra of matrices–addition, subtraction, and multiplication of matrices, determinant of a matrix, symmetric and skew-symmetric matrices, orthogonal matrix, rank of a matrix, inverse of a matrix, applications of matrices to solve system of linear equations,	11	**0**	**0**
IV	**Elementary Graph Theory:** • Basic terminologies of graphs, connected and disconnected graphs, subgraph, paths and cycles, complete graphs, digraphs, weighted graphs, Euler and Hamiltonian graphs. • Trees, properties of trees, concept of spanning tree. Planar graphs. Definitions and basic results on the topics mentioned.	**12**	**0**	**0**
Total		**45**	**0**	**0**

Definitions, examples, theorem statements and their applications, as well as practical uses of sets, relations, functions, counting, recurrence relations, graph theory, and matrix algebra.

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

1. **Set & Relation Assignment** – Solve problems using Venn diagrams and relation matrices.
2. **Counting Quiz** – Timed quiz on permutations, combinations, and pigeonhole principle.
3. **Graph Theory Mini-Project** – Draw and analyze simple graphs, trees, and networks.
4. **Matrix Lab** – Perform matrix operations and solve linear equations using a calculator or software.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1: Students will be able to remember few basics concepts needed to represent a graph.

CLO2. Students will be able to understand the learned mathematical foundations

CLO3: students will be able to apply the learned mathematical foundations in some standard situations

CLO4: Students will be able to analyse a given situation in light of introduced mathematical foundations.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge	CLO1	CLO2				
Conceptual Knowledge	CLO1	CLO2	CLO3	CLO4		
Procedural Knowledge			CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	0	0	0	0	0	0	0	0
CLO2	3	2	0	0	0	0	0	0	0	0
CLO3	3	2	0	0	0	0	0	0	0	0
CLO4	3	2	0	0	0	0	0	0	0	0
Average Contribution	3	2	0	0	0	0	0	0	0	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Garg, R. (2017). *Engineering Mathematics-I*. KHANNA PUBLISHING HOUSE.
2. Kolman, B., Busby, R. C., & Ross, S. (1995). *Discrete mathematical structures*. Prentice-Hall, Inc.
3. Narsingh, D. (1974). *Graph theory with applications to engineering and computer science*. Prentice-Hall.
4. Vasishtha, A. R., & Vasishtha, A. K. (1991). *Matrices*. Krishna Prakashan Media.
5. Garg, R. (2017). *Engineering Mathematics-I*. KHANNA PUBLISHING HOUSE.

Reference Books

1. Grimaldi, R. P. (2006). *Discrete and combinatorial mathematics, 5/e*. Pearson Education India.
2. Rosen, K. H., &Krithivasan, K. (1999). *Discrete mathematics and its applications* (Vol. 6). New York: McGraw-Hill.
3. West, D. B. (2001). *Introduction to graph theory* (Vol. 2). Upper Saddle River: Prentice hall.

Semester-I

Course Code : CC-5-BCA-102
Course Title : Computer Architecture
Nature of the Course : CORE
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives:

Objective-1. Understand number systems, binary arithmetic, and codes.
Objective -2. Learn to design and simplify digital circuits using Boolean logic.
Objective -3. Explain basic computer organization and control units.
Objective -4. Describe I/O and memory organization in computers.

Course Content:

Unit	Content	L	P	T
I	**Number Systems:** • Decimal, Binary, Octal, Hexadecimal, Number System Conversions, Binary Arithmetic, Addition and subtraction of BCD, Octal Arithmetic, Hexadecimal Arithmetic, Binary Codes, Decimal Codes, ASCII, EBCDIC, Excess-3 Code, The Gray Code. **Digital Principles:** • Basic concepts of digital logic design • Boolean algebra principles and logic gates • Logic gate symbols, truth tables, and Boolean expressions • Simplifying Boolean expressions using Karnaugh Maps (K-Maps), Don't Care Conditions, SOP and POS.	12	**14**	**0**
II	**Combinational Circuits:** • Half Adder and Full Adder, Subtractor, Decoders, Encoder, Multiplexer, Demultiplexer **Sequential Circuits:** • Flip-Flops- SR Flip- Flop, D Flip-Flop, J-K Flip-Flop, T Flip-Flop. **Register:** • 4 bit register with parallel load, Shift Registers, Bidirectional shift register **Binary Counters** • 4 bit synchronous and Asynchronous binary counter.	**10**	**18**	**0**
III	**Basic Computer Organization and Design:** • CPU registers and memory references • Instruction set architecture and addressing modes • Bus architecture and interconnection structures • Instruction cycle and execution • Control unit, hardwired control and microprogrammed control	11	**12**	**0**

	• RISC and CISC architectures			
IV	**Input-Output Organization:** • Peripheral Devices, Input-Output Interface, Modes of Transfer, Program controlled I/O, Interrupts, Direct memory Access, **Memory Organization:** • Memory Hierarchy, Main Memory, Auxiliary memory, Cache Memory and mapping techniques, cache replacement algorithms.	**12**	**16**	**0**
Total		**45**	**60**	**0**

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- **Number System Conversion Task** – Practice decimal, binary, octal, and hexadecimal conversions.
- **Logic Circuit Design** – Design and simplify circuits using Boolean algebra and Karnaugh Maps.
- **Flip-Flop and Counter Simulation** – Use simulation tools to demonstrate flip-flops and counters.
- **Computer Architecture Quiz** – Conduct a quiz on CPU organization, instruction cycles, and memory hierarchy.

Suggested list of LAB Assignments (not limited to):

1. Verify logic behavior of AND, OR, NAND, NOR, EX-OR, EX-NOR, Invert and Buffer gates.
2. To study and verify NAND as a Universal Gate
3. To verify De- Morgan's theorem for 2 variables
4. Design and test of an S-R flip-flop using NAND/NOR gate.
5. Convert BCD to Excess-3 code using NAND gate
6. To Convert Binary to Grey Code
7. Verification of Truth Tables of J-K Flip-Flop using NAND/NOR gate
8. Realize Decoder and Encoder circuit using Basic Gates.
9. Design and implement the 4:1 MUX using gates.
10. Implementation of 4-Bit Parallel Adder Using 7483 IC.
11. Design and verify operation of half adder and full adder.
12. Design and verify operation of half subtractor.
13. Design and Implement a 4 bit shift register using Flip flops.
14. Implement Boolean function using logic gates in both SOP and POS
15. Design and Implement a 4 bit synchronous counter.
16. Design and verify 4 bit asynchronous counter.

Hardware

1. Familiarize the computer system layout: marking positions of SMPS, motherboard, FDD, HDD, CD, DVD and add on cards.
2. Identify the Computer Name and Hardware Specification (RAM capacity, Processor type, HDD, 32 bit/ 64 bit)
3. Identify and Troubleshoot the problems of RAM, SMPS and motherboard
4. Configure BIOS settings- disable and enable USB and LAN
5. Adding additional RAM to the system.(expanding RAM size).
6. To Study mother board layout of a system.
7. Demonstrate the assembly of a PC
8. Demonstration of various ports: CPU, VGA port, PS/2 (keyboard, mouse) ,USB, LAN, Speaker, Audio.
9. Install and configure LINUX/windows OS
10. To study the installation of Printer and trouble shooting.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Convert between number systems and perform binary arithmetic.
CLO2. Design and simplify digital circuits using logic gates and Boolean expressions.
CLO3. Explain the working of computer components and instruction cycles.
CLO4. Describe input-output operations and memory hierarchy in computers.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO3, CLO4				
Conceptual Knowledge			CLO1			
Procedural Knowledge						CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	0	0	0	0	0	1	2	0
CLO2	3	2	0	0	0	0	0	2	0	0
CLO3	3	3	2	0	0	0	2	2	0	0
CLO4	3	0	0	0	0	1	0	1	2	0
Average Contribution	3	1.75	0.5	0	0	0.25	0.5	1.5	1	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books:

1. Donald, P. L., ALBERT, P. M., GOUTAM. SAHA, Malvino, A. P., & Saha, G. K. (2010). *Digital Principles & Applications*. Tata McGraw-Hill.
2. Mano, M. M. (1993). *Computer system architecture*. Prentice-Hall, Inc..

Reference Books:

1 Stallings, W. (2016). *Computer organization and architecture: designing for performance*. Pearson Education India.
2 Tanenbaum, A. S. (2016). *Structured computer organization*. Pearson Education India.
3 Subramanyam, M. V. (2005). *Switching theory and logic design*. Firewall Media.

SEMESTER–I

CC-2-BCA-103: Entrepreneurship and Startup Ecosystem

Course Code : CC-2-BCA-103
Course Title : Entrepreneurship and Startup Ecosystem
Nature of the Course : CORE
Total Credit assigned : 2
Component of the Course : Lecture: 1 Practical: 0 Tutorial: 1

Course Objective(s):

Objective-1. To introduce the fundamental concepts of entrepreneurship, its role in economic development, and provide insights into the characteristics and dynamics of Indian family businesses.

Objective-2. To equip students with the ability to identify viable business ideas and conduct effective feasibility and environmental analysis for new ventures.

Objective-3. To provide students with the practical knowledge required for setting up and managing new ventures, including marketing, team formation, funding, and legal aspects.

Objective-4. o familiarize students with the components of the start-up ecosystem and the support mechanisms available for entrepreneurs in India.

Course Content:

Unit	Content	P	T
I	**Introduction to Entrepreneurship & Family Business** • Definition and Concept of entrepreneurship • Entrepreneur Characteristics • Classification of Entrepreneurs • Role of Entrepreneurship in Economic Development–Start-ups • Knowing the characteristics of Family business with discussion on few Indian cases of Family Business like Murugappa, Dabur, Wadia, Godrej, Kirloskar etc.	0	3
II	Evaluating Business opportunity • Sources of business ideas and opportunity recognition • Guesstimating the market potential of a business idea • Feasibility analysis of the idea • Industry, competition and environment analysis	0	4
III	Building Blocks of starting ventures • Low cost Marketing using digital technologies • Team building from scratch • Venture Funding • Establishing the value-chain and managing operations • Legal aspects like IPR and compliances	0	4

IV	Start-up Ecosystem • Know the components of the start-up ecosystem including Incubators, Accelerators, Venture Capital Funds, Angel Investors etc. • Know various govt.schemes like Start-up India, Digital India, MSME etc. • Sources of Venture Funding available in India • Source of Technology, Intellectual Property management	0	4
		0	15

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

1. **Business Idea Pitch:**Students identify a business idea, estimate its market potential, and present it as a pitch to their peers.
2. **Family Business Case Study:**Analyze and discuss case studies of successful family businesses like Murugappa, Dabur, or Godrej to understand their operations and growth strategies.
3. **Feasibility Analysis Exercise:**Students conduct a feasibility analysis of a given business idea, considering market demand, competition, and financial viability.
4. **Low-Cost Marketing Plan:**Develop a low-cost marketing strategy using digital platforms for a startup idea, focusing on social media, SEO, and content marketing.
5. **Venture Funding Simulation:**Simulate a venture funding pitch where students present their business ideas to "investors" (classmates) for funding, practicing negotiation and pitching skills.
6. **Government Schemes Exploration:**Students research and present various government schemes for startups, such as Start-up India and MSME, and how they can benefit new ventures.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Define entrepreneurship, understand entrepreneurial traits, and analyze the structure and significance of family businesses in India.

CLO2. Evaluate the potential of a business idea using market, competition, and feasibility analyses.

CLO3. Design basic business operations including marketing strategies, funding sources, legal compliance, and team management for a start-up.

CLO4. Identify key players in the start-up ecosystem and explain government initiatives and funding sources that support entrepreneurship.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO3				
Conceptual Knowledge		CLO1, CLO4				
Procedural Knowledge						CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	0	0	0	0	0	1	2	0
CLO2	3	2	0	0	0	0	0	2	0	0
CLO3	3	3	2	0	0	0	2	2	0	0
CLO4	3	0	0	0	0	1	0	1	2	0
Average Contribution	3	1.75	0.5	0	0	0.25	0.5	1.5	1	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books :

1. *Startup India Leaning Program* by StartUp India available at www.startupindia.gov.in
2. Singh, A. (2021). Entrepreneurship by Rajeev Roy, Oxford University Press, India, 2020, Pages: 504, Rs. 799/-. *Abhigyan*, *39*(3), 51.
3. Barringer, B. R. (2008). *Entrepreneurship: Successfully launching new ventures*. Pearson Education India.
4. Agarwal, R. (2023). *Managing Family Business: Dynamics, Challenges, and Opportunities*. Routledge India.
5. Tiwari, A. (2023). Mapping the start-up ecosystem in India. *Economic and Political Weekly (Engage)*, *58*(32).

Semester-I

Course Code : SEC–5-BCA-101
Course Title : **Problem Solving Techniques**
Nature of the Course : **SEC**
Total Credit assigned : **5**
Component of the Course : **Lecture: 3 Practical: 2 Tutorial: 0**

Course Objectives

Objective 1: students will be able to remember about basic components of problem solving constructs such as flowcharts, algorithm, Pseudocode, program

Objective 2: students will be able to understand the importance of the programming constructs introduced to them.

Objective 3: Students will be able to apply the introduced program solving constructs on some previously encountered problem.

Objective 4: Students will be able to analyse the dependencies of problem solving constructs upto a program through the use of C language

Course Content:

Unit	Content	L	P	T
I	Problems And Problem Instances, Generalization and Special Cases, Types of Computational Problems, Classification of Problems, Analysis of Problems, Solution Approaches, Algorithm Development, Analysis of Algorithm, Efficiency, Correctness, Role of Data Structures inProblem Solving, Problem-Solving Steps (Understand the Problem, Plan, Execute, And Review), Breaking the Problem into Subproblems, Input/Output Specification, Input Validation, Pre and Post Conditions.	10	0	0
II	Structured Programming Concepts: Sequence (Input /Output / Assignment), Selection (If, If-Else) And Repetition (For, While, Do-While) Statements, Control Structure Stacking and Nesting. Different Kinds of Repetitions: Entry Controlled, Exit Controlled, Counter Controlled, Definite, Indefinite and Sentinel-Controlled Repetitions. Pseudocode and Flowcharts. Definition And Characteristics of Algorithms, Standard Algorithm Format. Problems Involving Iteration and Nesting: Displaying Different Patterns and Shapes Using Symbols and Numbers, Generating Arithmetic and Geometric Progression, Fibonacci and Other Sequences, Approximate Values For π, Sin(x), Cos(x), Etc. Using Taylor Series. Different Kinds of Data in The Real World and How They are Represented in The Computer Memory. Representation of Integers: Signed Magnitude Form, 1's Complement And 2's	15	20	0

	Complement. Representation of Real Numbers: IEEE 754 Floating Point Representation. Representation of Characters: ASCII, UNICODE. C Language: Introduction To Programming Languages, Different Generations of Programming Languages. Typed Vs Typeless Programming Languages, History of C Language, An Empty C Program. C Language Counterparts For Input (scanf()), Output (printf()) Statements, Assignment, Arithmetic, Relational and Logical Operators. If, If-Else Statements, For, While, Do-While Statements. Data Types. Translating Pseudocode/Algorithm to C Program. Incremental Compilation and Testing of The C Program. Simple Problems Involving Input, Output, Assignment Statement, Selection and Repetition. Good Coding Practices.			
III	Problems on Numbers: Extracting Digits of a Number (Left to Right and Right to Left), Palindrome, Prime Number, Prime Factors, Amicable Number, Perfect Number, Armstrong Number, Factorial, Converting Number from One Base to Another. Statistics (Maximum, Minimum, Sum and Average) on a Sequence of Numbers which are Read using Sentinel- Controlled Repetition using only a few Variables. C Language: else-if Ladder, switch Case, Increment/Decrement Operators, break and continue Statements.	10	20	0
IV	Modular Programming, Top-Down and Bottom-Up Approaches to Problem Solving. Recursion. Problems on Arrays: Reading and Writing of Array Elements, Maximum, Minimum, Sum, Average, Median and Mode. Sequential And Binary Search. Any one Sorting Algorithm. Matrix Operations. C Language: Function Definition and Declaration (Prototype), Role of Return Statement, One Dimensional and Two-Dimensional Arrays. String Functions. Other Operators, Operator Precedence and Associativity. Debugging.	10	20	0
Total		45	60	0

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

1. **Sessional activity:**

- Write algorithms and draw flowcharts for simple problems.
- Implement basic C programs using selection and repetition statements.
- Solve problems on number manipulation (palindrome, prime, factorial).
- Write C programs using arrays, functions, and recursion.

Suggested list of LAB Assignments (not limited to):

1. Converting degrees Celsius to Fahrenheit and vice versa?
2. Display three input numbers in sorted (non-decreasing) order?
3. Given a positive integer value n (>= 0) display number, square and cube of numbers from 1 to n in a tabular format?
4. Given an input positive integer number, display odd numbers from in the range [1,n]

5. Display first mathematical tables, each table up to 10 rows? Generalise this to display first n (> 0) mathematical tables up to m (m > 0) rows.
6. Given the first term (a), difference/multiplier (d) and number of terms (n > 0), display the first n terms of the arithmetic/geometric progression?
7. Display the first n (n > 0) terms of the fibonacci sequence.
8. Display the first n (n > 0) terms of the Tribonacci sequence?
9. Given two positive integer numbers n1 and n2 check if the numbers are consecutive numbers of the fibonacci sequence?
10. Write a program to calculate the sum and product of digits of an integer number.
11. Write a program to reverse a number.
12. Write a program to compute the sum of the first n terms of the series S = 1+1/2+1/3+1/4+……
13. Write a program to compute the sum of the first n terms of the series S =1-2+3-4+5……
14. Write a program that checks whether a given string is Palindrome or not.
15. Write a program to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
16. Write a program to compute the factors of a given number.
17. Write a program to swap two numbers using macros.
18. Write a program to perform various operations on an array entered by the user.
19. Write a program to print a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
20. Write a program to swap two numbers using pointers.
21. Write a program in which a function is passed address of two variables and then alter its contents.
22. Write a program which takes the radius of a circle as input from the user and computes its area and circumference.
23. Write a program to find sum of n elements entered by the user using dynamic memory allocation.
24. Write a program to perform various operations on strings using menu-driven approach.
25. Write a program to merge two ordered arrays of integers to get an ordered array.
26. Write a program to display Fibonacci series using recursion and iteration.
27. Write a program to calculate Factorial of a number using recursion and iteration.
28. Write a program to calculate GCD of two numbers using recursion and iteration.
29. Write a program to implement Matrix operations using templates.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Develop algorithms and represent them using flowcharts and pseudocode.

CLO2. Write C programs using control structures like if-else, loops, and switch-case.

CLO3. Solve problems involving numbers and sequences using C programming.

CLO4. Implement modular programming concepts with functions, arrays, and recursion in C.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO1				

Conceptual Knowledge						
Procedural Knowledge			CLO2, CLO3, CLO4			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	3	0	0	0	0	0	0	0
CLO2	2	0	2	0	0	1	0	0	0	0
CLO3	2	2	2	2	0	0	0	0	0	0
CLO4	2	0	3	3	0	0	0	0	0	0
Average Contribution	2.25	1	2.5	1.25	0	0.25	0	0	0	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Balagurusamy, E. (2018). *Computing fundamentals and C Programming*. McGraw-Hill Education.
2. Deitel, P. J., Deitel, H. M., & Deitel, A. (2013). *C: how to program*. Pearson..
3. Dromey, R. G. (1982). *How to Solve it by Computer*. Prentice-Hall, Inc..

Reference Books

1. Kernighan, B. W., & Ritchie, D. M. (1988). *The C programming language*. prentice-Hall.
2. Hanly, J. R., & Koffman, E. B. (2007). *Problem solving and program design in C*. Pearson Education India..

Semester-I

Indian Mathematics and Astronomy OLD

SEMESTER I

MDE-2-BCA-101: Indian Knowledge System in Computer Science (NEW)

Course Code : MDE-2-BCA-101
Course Title : Indian Knowledge System in Computing
Nature of the Course : Multidisciplinary Elective / IKS
Total Credit assigned : 2
Component of the Course : Lecture: 2 Practical: 0 Tutorial: 0

Course Objectives:

The main objectives of this course are:

Objective-1. To introduce students to selected Indian Knowledge System contributions that are relevant to computing and computer science.

Objective -2. To explain early ideas of counting, binary representation, combinatorial listing, and metrical patterns in the Indian intellectual tradition.

Objective -3. To relate the Katapayadi system to coding, symbolic mapping, mnemonic representation, and hashing-like ideas.

Objective -4. To discuss Paninian grammar, formal language representation, and automata-related perspectives in the history of computational thought.

Course Content

Unit	Content	L	P	T
I	**Indian Knowledge Systems and Computing Ideas** • Meaning and scope of Indian Knowledge System in relation to computing. • Overview of selected Indian scientific traditions: mathematics, grammar, logic, astronomy and cognition. • Computing science themes in the book: number representation, symbolic coding, formal grammar, automata and logic. • Need for critical, evidence-based and non-mythological study of historical scientific ideas.	6	0	0
II	**Number Representation, Binary Ideas and Combinatorial Thinking** • Pingala, Chandasastra and the classification of metrical patterns. • Laghu and guru syllables as two-symbol representation. • Prastara as systematic listing/enumeration of patterns. • Binary-number analogy and relevance to modern digital representation. • Basic combinatorial thinking: arrangement, sequencing and pattern generation.	8	0	0

Unit	Content	L	P	T
III	**Katapayadi System, Symbolic Coding and Hashing Analogy** • Katapayadi system as a method of assigning numerical values to letters. • Encoding of words, numbers and mnemonic representations. • Comparison with key-value mapping and hashing-like storage ideas. • Introductory discussion on hashing: keys, positions, retrieval and collision as conceptual comparison only. • Examples such as verse-based numerical encoding and raga classification as pattern organisation.	8	0	0
IV	**Formal Grammar, Automata and Indian Logic** • Paninian grammar as a rule-based formal system. • Basic idea of production rules, syntax, grammar and Backus-Naur Form comparison. • Fowler's automaton-related discussion and grammar testing as historical-computational perspective. • Formal structures in Indian logic: propositions, inference, validity and rule-based reasoning. • Relevance to formal languages, compiler design, natural language processing and artificial intelligence.	8	0	0
Total		**30**	**0**	**0**

Mode of In-Semester Assessment

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activity

- Short note on the relevance of Indian Knowledge Systems to computing and computational thinking.
- Seminar presentation on Pingala's metrical patterns and binary-number analogy.
- Small assignment on Katapayadi as a symbolic coding system and its comparison with hashing or key-value mapping.
- Comparative note on Paninian grammar, Backus-Naur Form and formal grammar in programming languages.
- Book review / reflective report based on selected chapters of Computing Science in Ancient India.

Evaluation Criteria

- Conceptual clarity and historical accuracy.
- Ability to connect IKS concepts with modern computer-science ideas.
- Quality of examples, presentation and written explanation.
- Critical and evidence-based interpretation of source material.

Suggested Assignment / Seminar Work

- Prepare a table showing binary patterns using laghu/guru or 0/1 notation for simple metrical combinations.

- Write a short report on how the Katapayadi scheme may be compared with symbolic encoding and hashing.
- Draw a simple grammar-rule example and compare it with Paninian rule-based grammar or Backus-Naur Form.
- Explain one example of rule-based reasoning from Indian logic and relate it to inference in computing.
- Prepare a brief presentation on the importance of studying historical computing ideas critically.

Expected Course Learning Outcomes

By the end of the course, students will be able to:

CLO1. Explain selected Indian Knowledge System contributions relevant to computing and computational thinking.

CLO2. Interpret binary-like representation, metrical pattern generation and combinatorial listing in selected ancient Indian sources.

CLO3. Relate the Katapayadi system to symbolic coding, mnemonic representation and hashing-like conceptual mapping.

CLO4. Describe the relevance of Paninian grammar, formal rules, automata-related discussion and Indian logic to modern computer science.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO4				
Conceptual Knowledge		CLO1, CLO2, CLO3				
Procedural Knowledge						
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	0	0	0	0	0	1	0	0	2	0
CLO2	0	2	0	0	0	2	0	0	0	0
CLO3	0	0	0	0	0	2	0	0	0	0
CLO4	0	0	0	0	0	0	0	0	2	0
Average Contribution	**0**	**0.5**	**0**	**0**	**0**	**1.25**	**0**	**0**	**1**	**0**

(3 = High, 2 = Moderate, 1 = Low, - = No direct correlation; average contribution treats "-" as 0.)

Suggested Readings

1. Rao, T. R. N., & Kak, S. (Eds.). (1995). Computing science in ancient India. Center for Advanced Computer Studies, University of Southwestern Louisiana.

Semester-II

Course Code :CC-3-BCA-201
Course Title : Mathematics Foundations to Computer Science - II
Nature of the Course : CORE
Total Credit assigned : 3
Component of the Course : Lecture: 3 Practical: 0 Tutorial: 0

Course Objectives:

Objective-1. Understand and apply principles of logic and proof techniques in problem solving.
Objective--2. Explain basic algebraic structures including groups and subgroups.
Objective--3. Solve numerical methods problems using root finding, interpolation, and integration techniques.
Objective--4. Formulate and solve optimization problems using linear programming and transportation methods.

Course Content:

Unit	Content	L	P	T
I	**Logic and Methods of Proofs:** Propositions, logical operations (basic connectives), compound statements, construction of truth table, quantifiers, conditional statements, tautology, contradiction, contingency, logical equivalence. Conjunctive Normal Forms (CNF) and Disjunctive Normal Forms (DNF). Methods of proofs: Rules of inference for propositional logic, modus ponens, modus tollens, syllogism, proof by contradiction, Mathematical Induction.	13	**0**	**0**
II	**Algebraic Structures:** Semi-group, Monoid, Group, Subgroup, Cyclic group. *Definitions, examples, theorem statements and their applications, as well as practical uses.*	**12**	**0**	**0**
III	**Numerical Methods:** Concept and importance of errors in numerical methods. Solution of algebraic and transcendental equations: Bisection method and Newton-Raphson methods. Numerical Interpolation: Newton's Forward and Newton's Backward interpolation formula and Lagrange's formula. Numerical Integration: Trapezoidal rule and Simpson's 1/3 rule *Only formula and problem solving for all the topics mentioned above.*	20	**0**	**0**
Total		**45**	**0**	**0**

Mode of In-Semester assessment:

A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Construct truth tables and prove logical equivalences using inference rules.
- Solve problems on algebraic structures like groups and subgroups.
- Apply numerical methods: implement Bisection, Newton-Raphson, and interpolation formulas.
- Formulate and solve linear programming and transportation problems using graphical and simplex methods.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Construct and analyze logical statements and proof methods.

CLO2. Explain and solve problems related to algebraic structures.

CLO3. Apply numerical methods for root finding, interpolation, and integration.

CLO4. Formulate and solve optimization problems using linear programming techniques.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO2				
Conceptual Knowledge				CLO1		
Procedural Knowledge			CLO3			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	0	0	0	0	0	0	0	0
CLO2	3	2	0	0	0	1	0	0	2	0
CLO3	3	2	0	0	0	0	0	0	0	0
CLO4	3	3	2	2	0	0	0	0	0	0
Average Contribution	3	2.25	0.5	0.5	0	0.25	0	0	0.5	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Kolman, B., Busby, R. C., & Ross, S. (1995). *Discrete mathematical structures*. Prentice-Hall, Inc..
2. Sastry, S. S. (2012). *Introductory methods of numerical analysis*. PHI Learning Pvt. Ltd..
3. Taha, H. A. (2013). *Operations research: an introduction*. Pearson Education India.

Reference Books

1. Rosen, K. H., &Krithivasan, K. (1999). *Discrete mathematics and its applications* (Vol. 6).

New York: McGraw-Hill.

2. Sharma, J. K. (2009). *Operations research theory and application*. MACMILAN Publishers.

Semester-II

CC-5-BCA-202: Data Structure

Course Code **:CC-5-BCA-202**
Course Title **:** Data Structure
Nature of the Course **: CORE**
Total Credit assigned **: 5**
Component of the Course **: Lecture: 3 Practical: 2 Tutorial: 0**

Course Objectives:

Objective-1. To introduce students to fundamental data structures and array operations, and to develop understanding of algorithm complexity and efficiency.
Objective -2. To develop skills in dynamic memory data structures like linked lists and understand efficient data retrieval using hashing techniques.
Objective -3. To provide conceptual and practical knowledge of stack and queue data structures along with recursive problem-solving techniques.
Objective -4. To understand non-linear data structures like trees and graphs and apply them for efficient data organization and traversal.

Prerequisite:

1. **Programming Fundamentals:** Understanding the basic syntax and semantics of C programming language.
2. **Problem-Solving Skills:** Ability to break down a problem into smaller steps and devise a step-by-step solution and familiarity with simple algorithms.

Course Content:

Unit	Content	L	P	T
I	**Introduction and Overview:** • Definition, Classification and Operations of Data Structures. • Algorithms: Complexity, Time-Space Tradeoff. **Arrays:** • Definition and Classification of Arrays, Representation of Linear Arrays in Memory, • Operations on Linear Arrays: Traversing, Inserting, Deleting, Searching, Sorting and Merging. • Searching: Linear Search and Binary Search, Comparison of Methods. Sorting: Bubble Sort, • Selection Sort, and Insertion Sort. Two-Dimensional Arrays, • Representation of Two-Dimensional Arrays in Memory, Matrices and Sparse Matrices, Multi-Dimensional Arrays.	10	**14**	**0**
II	**Linked Lists:** • Definition, Comparison with Arrays, Representation, Types of Linked lists, Traversing, Inserting, Deleting and Searching in Singly Linked List, Doubly Linked List and Circular Linked List. Applications of Linked Lists: Addition of Polynomials. **Hashing and Collision:** • Hashing, Hash Tables, Types of Hash Functions, Collision, Collision Resolution with Open Addressing and Chaining.	**12**	**18**	**0**

III	**Stacks:** Definition, Representation of Stacks using Arrays and Linked List, Operations on Stacks using Arrays and Linked List, Application of Stacks: Arithmetic Expressions, Polish Notation, Conversion of Infix Expression to Postfix Expression, Evaluation of Postfix Expression. **Recursion:** Definition, Recursive Notation, Runtime Stack, Applications of Recursion: Factorial of Number, GCD, Fibonacci Series and Towers of Hanoi. **Queues:** Definition, Representation of Queues using Array and Linked List, Types of Queue: Simple Queue, Circular Queue, Double-Ended queue, Priority Queue, Operations on Simple Queues and Circular Queues using Array and Linked List, Applications of Queues.	13	**16**	**0**
IV	**Graphs:** Definition, Terminology, Representation, Traversal. **Trees:** Definition, Terminology, Binary Trees, Traversal of Binary Tree, Binary Search Tree, Inserting, Deleting and Searching in Binary Search Tree, Height Balanced Trees: AVL Trees, Insertion and Deletion in AVL Tree.	**10**	**12**	**0**
Total		**45**	**60**	**0**

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Implement and demonstrate operations on arrays and linked lists.
- Write programs to perform stack and queue operations, including expression evaluation.
- Solve problems using recursion (factorial, Fibonacci, Tower of Hanoi).
- Implement basic graph and tree traversals and operations on binary search trees and AVL trees.

Suggested list of LAB Assignments (not limited to):

1. Implement and analyse the time complexity of various operations on an array, such as insertion, deletion, and searching.
2. Implement a singly linked list and perform operations such as insertion, deletion, and searching.
3. Implement a stack using an array and perform stack operations, such as push, pop, and peek. Analyse the time complexity of these operations.
4. Implement a queue using an array and perform queue operations, such as enqueue, dequeue, and peek. Analyse the time complexity of these operations.
5. Implement the conversion of infix expressions to postfix using a stack.
6. Implement a binary search tree and perform operations such as insertion, deletion, and searching.
7. Implement a heap and perform heap operations, such as insert, delete, and build heap.
8. Implement various sorting algorithms, such as bubble sort, selection sort, insertion sort, quicksort, and mergesort, and analyse their time complexity.
9. Implement a hash table using separate chaining collision resolution and analyse its time complexity.

10. Implement dynamic hashing and analyse its performance.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Classify data structures, analyze algorithm efficiency, and implement linear and multi-dimensional arrays with basic operations and sorting/searching techniques.
CLO2. Construct and manipulate various types of linked lists, apply them to real-world problems, and implement hash tables with collision resolution strategies.
CLO3. Implement stack and queue operations using arrays and linked lists, solve expression-based problems using stacks, and apply recursion to solve mathematical problems.
CLO4. Construct and traverse trees and graphs, implement binary search trees and AVL trees, and perform efficient insertion and deletion operations.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge				CLO1		
Procedural Knowledge			CLO3			CLO2, CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	3	0	0	0	0	0	1	0
CLO2	3	2	3	0	0	0	0	0	1	0
CLO3	3	2	3	0	0	0	0	0	1	0
CLO4	3	2	3	0	0	0	0	0	1	0
Average Contribution	3	2	3	0	0	0	0	0	1	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Patel, R. B. (2001). *Expert Data Structure with C*. KHANNA PUBLISHING HOUSE. (AICTE Recommended Textbook)
2. Seymour, L., &VijayalakshmiPai, G. A. (2010). Data Structures with C.
3. Kanetkar, Y. (2019). *Data Structures Through C: Learn the fundamentals of Data Structures through C*. BPB publications.

Reference Books

1. Thareja, R. (2014). Data structures using C..
2. Horowitz, E., & Sahni, S. (1982). Fundamentals of data structures..

Semester-II

Course Code :**CC-3-BCA-203**
Course Title :**Probability and Statistics**
Nature of the Course :**CORE**
Total Credit assigned : **3**
Component of the Course : **Lecture: 3 Practical: 0 Tutorial: 0**

Course Objectives:

Objective-1. To provide a foundational understanding of statistical data types, data presentation, measures of central tendency and dispersion, and basic probability theory.
Objective -2. To introduce the concept of random variables and their distributions, along with standard probability distributions such as Binomial, Poisson, and Normal.
Objective -3. To explain the concepts of correlation and regression, and enable learners to assess relationships between variables using statistical methods.
Objective -4. To develop understanding of sampling concepts and basic inferential statistics, including hypothesis testing for means, proportions, and categorical data.

Course Content

Unit	Content	L	P	T
I	Basic concepts of Statistics, qualitative and quantitative data, classification of data, construction of frequency distribution, diagrammatic representation of data. • **Measures of Central Tendency**: Arithmetic mean, median and mode—their properties • **Measures of Dispersion**: Range, mean deviation, quartile deviation, variance and standard deviation. • **Concepts of probability:** Experiment and sample space, events and operations with events, probability of an event, basic probability rules, applications of probability rules, conditional probability	13	**0**	**0**
II	• **Random Variables:** Discrete and continuous random variable**,** probability distribution of a random variable**,** probability mass function, probability density function, expectation and variance of a random variable. • **Standard Probability Distributions:** Binomial probability distribution, Poisson probability distribution, Normal probability distribution.	**11**	**0**	**0**
III	• **Correlation: Definition, scatter diagram, types of correlation, measures—Karl Pearson's correlation coefficient and Spearman's rank correlation coefficient.** • **Regression**: Linear regression-fitting by least square method and interpretation.	11	**0**	**0**

IV	• **Sampling Distribution:** Concept of Population and Sample, parameter and statistic, sampling distribution of sample mean and sample proportion. • **Statistical Inference:** Estimation and Hypothesis Testing (only concept). • **Hypothesis Testing for a Single Population:** Concept of a hypothesis testing, tests involving a population mean and population proportion (z test and t test). • **Chi square test for independence of attributes and goodness of fit**.	**10**	**0**	**0**
Total		**45**	**0**	**0**

(Definitions, examples, theorem statements and their applications, as well as practical uses.)

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- **Data Collection & Visualization:**
  - Students collect real-world data, organize it into frequency distributions, and create histograms or bar charts for visual representation.
- **Regression & Correlation Analysis:**
  - Using a provided dataset, students calculate correlation coefficients and perform linear regression to analyze relationships between variables.
- **Probability Simulation:**
  - Students simulate random experiments (e.g., coin tosses, dice rolls) to understand probability distributions and compare theoretical vs. experimental probabilities.
- **Hypothesis Testing Exercise:**
  - Students conduct z-tests or t-tests on real-world data to test hypotheses about population means or proportions and interpret results.
- **Chi-Square Test for Independence:**
  - Students analyze contingency tables and perform chi-square tests to assess the independence of categorical variables in a given dataset.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Classify data, construct frequency distributions, compute key statistical measures, and apply basic probability rules to simple problems.

CLO2. Analyze discrete and continuous random variables, compute expected values and variances, and apply standard probability distributions in practical contexts.

CLO3. Calculate correlation coefficients, interpret scatter plots, and perform linear regression analysis using the least squares method.

CLO4. Differentiate between population and sample, perform hypothesis tests (z, t, and chi-square), and interpret statistical inference results.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO1, CLO2, CLO4				
Conceptual Knowledge					CLO3	
Procedural Knowledge						
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	3	0	0	0	0	0	0	1	0
CLO2	3	2	0	0	0	0	0	0	0	0
CLO3	3	2	0	0	0	0	0	0	0	0
CLO4	3	3	0	0	0	0	0	0	0	0
Average Contribution	3	2.5	0	0	0	0	0	0	0.25	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Ross, S. (2009). Probability and statistics for engineers and scientists. *Elsevier, New Delhi*, *16*, 32-33.
2. Devore, J. L. (2020). Probability and statistics for engineering and the sciences (9th ed.). Cengage Learning.

Semester-II

Course Code :**SEC-5-BCA-201**
Course Title : **Object Oriented Programming using Java**
Nature of the Course : **SEC**
Total Credit assigned : **5**
Component of the Course : **Lecture: 3 Practical: 2 Tutorial: 0**

Course Objectives

Objective-1. To introduce the object oriented programming system concepts
Objective -2. To introduce syntax and semantics of Java programming language
Objective -3. To develop modular programs using Java
Objective -4. To setup JDK environment to create, debug and run Java programs

Prerequisite: Knowledge of Problem Solving Techniques using C programming language

Course Content:

Unit	Content	L	P	T
I	**Fundamentals of Object Oriented Programming:** Basic concepts of Object Oriented Programming (OOP), Benefits and Applications of OOP. **Java Evolution:** Java Features, Difference between Java, C and C++, Java and Internet, Java Environment. **Overview of Java Language:** Introduction to Simple Java Program, Use of Comments and Math function, Application of two classes, Java Program Structure, Java Tokens and statements, Implementing Java program And JVM, Command Line Arguments.	12	**10**	**0**
II	**Constants, Variables and Data Types:** Constants, Variables, Data Types, Declaration of Variables, Giving values to Variables, Symbolic Constants, Type casting. **Operators & Expressions:** Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment & Decrement operators, conditional operators, Bitwise operators, Arithmetic Expressions, Evaluation of Expressions, Type Conversions in Expressions, Operator Precedence & Associativity. **Decision Making, Branching & Looping:** Decision Making with Control Statements, Looping statements, Jump in loops, Labelled loops.	**13**	**16**	**0**
III	**Classes, Objects and Methods:** Defining Class, Methods Declaration, Constructors, Methods Overloading, Overriding Methods, Inheritance **Arrays, Strings and Vectors:** 1D arrays, Creating an Array, 2D arrays, Strings, Vectors, Wrapper Classes, Enumerated Types **Inheritance:** Defining, extending classes, and Implementing Interfaces. Multiple inheritance and polymorphism.	10	**16**	**0**

IV	**Packages:** Basics of packages, System packages, Creating and accessing packages, Creating user defined packages, Adding class to a package. **Exception Handling:** Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions	**10**	**18**	**0**
Total		**45**	**60**	**0**

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Write and run simple Java programs demonstrating basic syntax and console input/output.
- Implement programs using decision making and looping constructs (if-else, for, while).
- Create Java classes and objects with methods, constructors, and demonstrate method overloading/overriding.
- Develop Java programs using arrays, strings, inheritance, and interfaces.
- Write programs to handle exceptions using try-catch blocks and create custom exceptions.
- Create and use Java packages, including user-defined packages.

Suggested list of LAB Assignments (not limited to):

1. Write a program to read two numbers from user and print their product.
2. Write a program to print the square of a number passed through command line arguments.
3. Write a program to send the name and surname of a student through command line arguments and print a welcome message for the student.
4. Write a java program to find the largest number out of n natural numbers.
5. Write a java program to find the Fibonacci series & Factorial of a number using recursive and non recursive functions.
6. Write a java program to multiply two given matrices.
7. Write a Java program for sorting a given list of names in ascending order.
8. Write a Java program that checks whether a given string is a palindrome or not .Ex:MADAM is a palindrome.
9. Write a java program to read n number of values in an array and display it in reverse order.
10. Write a Java program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.
11. Create a JAVA class called Student with the following details as variables within it.
    a. USN, NAME, BRANCH, PHONE, PERCENTAGE
    b. Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentage of these objects with suitable headings.
12. Write a Java program that displays the number of characters, lines and words in a text.
13. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.
14. Write a Java program to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, override the calculateSalary() method to calculate and return the salary based on their specific roles.

15. Write a Java program using an interface called 'Bank' having function 'rate_of_interest()'. Implement this interface to create two separate bank classes 'SBI' and 'PNB' to print different rates of interest. Include additional member variables, constructors also in classes 'SBI' and 'PNB'.
16. Write a Java package program for the class book and then import the data from the package and display the result.
17. Write a Java program for finding the cube of a number using a package for various data types and then import it in another class and display the results.
18. Write a Java program for demonstrating the divide by zero exception handling.
19. Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
20. Create an exception subclass UnderAge, which prints "Under Age" along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.

Expected Course Learning Outcome:

By the end of the course, students will be able to:

CLO-1. Explain the fundamental concepts and benefits of Object-Oriented Programming and Java environment.
CLO-2. Demonstrate use of variables, data types, operators, and control statements in Java programs.
CLO-3. Design and implement classes, objects, methods, arrays, and inheritance concepts in Java.
CLO-4. Apply exception handling and package creation in Java programs.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO1				
Conceptual Knowledge		CLO2				
Procedural Knowledge						CLO3, CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	2	2	0	0	0	0	0	0
CLO2	2	0	2	2	0	0	0	0	0	0
CLO3	2	0	3	3	0	0	0	0	0	0
CLO4	2	2	3	2	2	0	0	0	0	3
Average Contribution	2	0.5	2.5	2.25	0.5	0	0	0	0	**0.75**

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Balaguruswamy, E. (2014). *Programming with Java-A Primer*. McGraw-Hill Professionals.
2. Schildt, H. (2022). Java™ The Complete Reference Twelfth Edition.

Reference Books

1. Liang, Y. D. (2015). *Introduction to Java programming* (Vol. 7). Boston: Pearson.
2. Malhotra, S., & Choudhary, S. (2011). *Programming in JAVA*. Oxford University Press, Inc..

Semester-II

Course Code :**SEC-2-BCA-202**
Course Title : **Web Technologies**
Nature of the Course : **SEC**
Total Credit assigned : **2**
Component of the Course : **Lecture: 1 Practical: 1 Tutorial: 0**

Course Objectives:

Objective-1. Understand and apply the fundamentals of web development including HTML, CSS, Bootstrap, and web hosting concepts.

Objective -2. Develop interactive web applications using JavaScript, AJAX, XML, and JSON for client-side scripting and data exchange.

Course Content:

Unit	Content	L	P	T
I	Introduction to HTML, history of HTML, Objective, basic Structures of HTML, Header Tags, body tags, Paragraph Tags. Tags for FORM Creation, TABLE, FORM, TEXTAREA, SELECT, IMG, IFRAME FIELDSET, ANCHOR. Lists in HTML, Introduction to DIV tag, NAVBAR Design. Introduction to CSS, types, Selectors, and Responsiveness of a web page. Introduction to Bootstrap, downloads/linking, using classes of Bootstrap, understanding the Grid System in Bootstrap. Introduction to www, Protocols and Programs, Applications and development tools, web browsers, DNS, Web hosting Provider, Setting up of Windows/Linux/Unix web servers, Web hosting in cloud, Types of Web Hosting.	7	**12**	**0**
II	Introduction to JavaScript: Functions and Events, Document Object model traversing using JavaScript. Output System in JavaScript i.e. Alert, throughput, Input box, Console. Variables and Arrays in JavaScript.Date and String handling in JavaScript. Manipulating CSS through JavaScript: Form Validation like Required validator, length validator, Pattern validator. Advanced JavaScript, Combining HTML, CSS and JavaScript events and buttons, controlling your browser.Introduction to AJAX, Purpose, advantages and disadvantages, AJAX based Web applications and alternatives of AJAX. Introduction to XML: uses, Key concepts, DTD 8 schemas, XSL, XSLT,and XSL Elements and transforming with XSLT. Introduction to XHTML. JSON: Introduction to JSON, Keys and Values, Types of Values, Arrays, Objects	**8**	**18**	**0**
Total		**15**	**30**	**0**

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Create a basic HTML webpage using different tags like headings, paragraphs, lists, tables, forms, images, and links.
- Design a responsive webpage using CSS and Bootstrap grid system.
- Implement JavaScript functions for DOM manipulation, event handling, and form validation.
- Build a simple web page combining HTML, CSS, and JavaScript with interactive elements (buttons, alerts, input validation).
- Demonstrate basic AJAX calls to fetch data asynchronously.
- Create and transform XML data using XSLT.
- Write and parse JSON data with JavaScript for dynamic content updates.

Suggested list of LAB Assignments (not limited to):

PART-A

1. Create your class time table using table tag.
2. Design a Webpage for your college containing description of courses, department, faculties, library etc. using list tags, href tags, and anchor tags.
3. Create web page using Frame with rows and columns where we will have header frame, left frame, right frame, and status bar frame. On clicking in the left frame, information should be displayed in right frame.Create
4. Your Resume using HTML, use text, link, size, color and lists.
5. Create a Web Page of a super market using (internal CSS)
6. Use Inline CSS to format your resume that you have created.
7. Use External CSS to format your time table created.
8. Use all the CSS (inline, internal andexternal) toformat college web page that you have created.
9. Write a HTML Program to create your college website using for mobile device.

PART – B

1. Write an HTML/JavaScript page to create login page with validations.
2. Develop a Simple calculator for addiction, subtraction, multiplication and division operation using JavaScript.
3. Use Regular Expressions for validations in Login Page using JavaScript.
4. Write a Program to retrieve date from a text file and displaying it using AJAX.
5. Create XML file to store Student Information like Register Number, Name, Mobile Number, DOB, and Email-Id.
6. Create a DTD for (0).
7. Create XML scheme for (0).
8. Create XSL file to convert XML file to XHTML file.
9. Write a JavaScript program using Switch case.
10. Write a JavaScript program using any 5 events.
11. Write a JavaScript program using built in JavaScript objects.
12. Write program for populating values from JSON text.
13. Write a program to transform JSON text to a JavaScript object.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Design and develop responsive static web pages using HTML, CSS, and Bootstrap, and demonstrate basic knowledge of web hosting and server setup.

CLO2. Implement interactive web functionalities using JavaScript and handle data exchange using AJAX, XML, and JSON effectively in web applications.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge			CLO1			CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	0	1	1	0	0	0	0	1	0
CLO2	3	2	3	3	0	0	0	0	1	0
CLO3	2	0	2	3	0	0	0	0	1	0
CLO4	2	0	1	1	0	1	0	1	2	0
Average Contribution	2.5	0.5	1.75	2	0	0.25	0	0.25	1.25	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Lemay, L., Colburn, R., &Kyrnin, J. (2016). Mastering HTML, CSS & JavaScript, Web Publishing.
2. Powell, T. A. (2004). *The Complete Reference: HTML & XHTML*. McGraw Hill.

Reference Books

1. Moreto, S. (2016). *Bootstrap 4 By Example*. Packt Publishing Ltd..

SEMESTER –III

Course Code :CC-4-BCA-301
Course Title : Operating Systems
Nature of the Course : CORE
Total Credit assigned : 4
Component of the Course : Lecture: 3 Practical: 1 Tutorial: 0

Course Outcomes:

Objective-1. Understand the fundamental concepts, components, and types of operating systems.
Objective-2. Explain process management, scheduling algorithms, and multithreading concepts.
Objective-3. Analyze process synchronization, deadlocks, and related prevention and recovery techniques.
Objective-4. Describe memory management, virtual memory, and I/O management including scheduling algorithms.

Course Content:

Unit	Content	L	P	T
I	**Operating Systems Overview:** Definition, Evaluation of O.S, Components & Services of OS, Structure, Architecture, types of Operating Systems, Batch Systems, Concepts of Multiprogramming and Time Sharing, Parallel, Distributed and real time Systems. **Operating Systems Structures:** Operating system services and systems calls, system programs, operating system structure, operating systems generations.	10	**6**	**0**
II	**Process Management:** Process Definition, Process states, Process State transitions, Process Scheduling, Process Control Block, Threads, Concept of multithreads, Benefits of threads, Types of threads. **Process Scheduling:** Definition, Scheduling objectives, Scheduling algorithms, CPU scheduling Preemptive and Non-preemptive Scheduling algorithms (FCFS, SJF and RR), Performance evaluation of the scheduling Algorithms	**12**	**8**	**0**
III	**Process Synchronization:** Introduction, Inter-process Communication, Race Conditions, Critical Section Problem, Mutual Exclusion, Semaphores, Monitors. **Deadlocks:** System model, deadlock characterization, deadlock prevention, avoidance, Banker's algorithm, Deadlock detection, and recovery from deadlocks.	13	**8**	**0**
IV	**Memory Management:** Logical and Physical address map, Swapping, Memory allocation, MFT, MVT, Internal and External fragmentation and Compaction, Paging, Segmentation. **Virtual Memory:** Demand paging, Page Replacement algorithms, Allocation of frames, thrashing. **I/O Management:** Principles of I/O Hardware: Disk structure, Disk scheduling algorithms.	**10**	**8**	**0**
Total		**45**	**30**	**0**

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Illustrate and explain different types of operating systems and their structures.
- Simulate CPU scheduling algorithms like FCFS, SJF, and Round Robin with example processes.
- Solve problems on process synchronization using semaphores and critical section solutions.
- Analyze deadlock scenarios and apply prevention or avoidance methods.
- Demonstrate memory management techniques like paging and segmentation through examples.
- Study and compare disk scheduling algorithms using sample requests.

Suggested list of LAB Assignments (not limited to):

1. Write C program to simulate the FCFS CPU Scheduling algorithm.
2. Write C program to simulate the SJF CPU Scheduling algorithm.
3. Write C program to simulate the Round Robin CPU Scheduling algorithm.
4. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
5. Write a C program to implement the Producer – Consumer problem using semaphores.
6. Write a C program to illustrate the IPC mechanism using Pipes.
7. Write a C program to illustrate the IPC mechanism using FIFOs.
8. Write a C program to simulate Paging memory management technique.
9. Write a C program to simulate Segmentation memory management technique.
10. Write a C program to simulate the Best Fit contiguous memory allocation technique.
11. Write a C program to simulate the First Fit contiguous memory allocation technique.
12. Write a C program to simulate the concept of Dining-Philosophers problem.
13. Write a C program to simulate the MVT algorithm.
14. Write a C program to implement FIFO page replacement technique.
15. Write a C program to write a C program for implementing sequential file allocation method.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Describe the structure and types of operating systems.

CLO2. Implement and analyze CPU scheduling algorithms.

CLO3. Apply synchronization techniques and handle deadlocks.

CLO4. Explain memory management and I/O scheduling methods.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create

Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge			CLO1			CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	0	1	1	0	0	0	0	1	0
CLO2	3	2	3	3	0	0	0	0	1	0
CLO3	2	0	2	3	0	0	0	0	1	0
CLO4	2	0	1	1	0	1	0	1	2	0
Average Contribution	2.5	0.5	1.75	2	0	0.25	0	0.25	1.25	**0**

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books:

1. Walia, E. (2002). *Operating system concepts*. KHANNA PUBLISHING HOUSE. (AICTE Recommended Textbook)
2. Silberschatz, A., Galvin, P. B., & Gagne, G. (2006). *Operating system principles*. John Wiley & Sons.
3. Stallings, W. (2018). *Operating Systems: Internals and Design Principles, 9/e*. Pearson IT Certification.

Reference Books:

1. Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems*. Pearson Education, Inc...

SEMESTER –III

CC-5-BCA-302: Database Management System

Course Code :**CC-5-BCA-302**
Course Title : **Database Management System**
Nature of the Course : **CORE**
Total Credit assigned : **5**
Component of the Course : **Lecture: 3 Practical: 2 Tutorial: 0**

Course Objectives:

Objective-1. To introduce the fundamental concepts of databases, data models, database users, and design using the ER model and relational schema.
Objective -2. To develop skills in querying databases using SQL and relational algebra, and to understand normalization techniques for effective database design.
Objective -3. To explore the principles of transaction management, concurrency control, recovery techniques, and database indexing and storage strategies.
Objective -4. To introduce NoSQL databases, big data technologies, and key concepts in database security and advanced data management.

Course Content

Unit	Content	L	P	T
I	• **Introduction to Databases**: Definition of Data, Database, and DBMS, Overview of Database Applications, Advantages and Disadvantages of DBMS, Roles of Database Users and Administrators • **Data Models:** Introduction to Data Models, Types of Data Models (Hierarchical, Network, Relational, Object-oriented), Importance of Data Models in DBMS • **Database Design**: Keys: Primary Key, Candidate Key, Super Key, Foreign Key, Composite Key, Alternate Key, Unique Key, Surrogate Key, Constraints in a table: Primary Key, Foreign Key, Unique Key, NOT NULL, CHECK, Entity-Relationship (ER) Model, Entities and Entity Sets, Attributes and Relationships, ER Diagrams, Key Constraints and Weak Entity Sets, Extended ER Features, Introduction to the Relational Model and Relational Schema	12	12	0
II	• **Relational Algebra and Calculus**: Introduction to Relational Algebra, Operations: Selection, Projection, Set Operations, Join Operations, Division, Tuple and Domain Relational Calculus • **Structured Query Language (SQL):** SQL Basics: DDL and DML, Aggregate Functions (Min(), Max(), Sum(), Avg(), Count()), Logical operators (AND, OR, NOT), Predicates (Like, Between, Alias, Distinct),	12	22	0

	Clauses(Group By, Having, Order by, top/limit), Inner Join, Natural Join, Full Outer Join, Left Outer Join, Right outer Join, Equi Join • **Normalization and Database Design: Functional Dependencies: Armstrong's Axioms, Definition, Properties (Reflexivity, Augmentation, Transitivity), Types (Trivial, Non-Trivial, Partial and Full Functional Dependency), Closure of Functional Dependencies, Normal Forms(1NF, 2NF, 3NF, BCNF),**			
III	• **Transaction Management**: ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control, Performance of Locking, Transaction Support in SQL, Introduction to Crash Recovery, 2PL, Serializability, and Recoverability, Introduction to Lock Management, Dealing with Deadlocks • **Database Storage and Indexing**: Data on External Storage, File Organizations and Indexing, Index Data Structures, Comparison of File Organizations, Indexes and Performance Tuning, Guidelines for Index Selection, Basic Examples of Index Selection	11	12	0
IV	• **NoSQL Databases and Big Data:** Introduction to NoSQL, Data Models: Document, Key value, Column family, Graph. Uses and Features of NO/SQL document databases. CAP theorem, BASE vs ACID, CRUD operations, Mongo DB operators, Overview of Big Data Technologies: Hadoop, MongoDB, Cassandra. • **Database Security and Advanced Topics:** Introduction to Database Security, Access Control, Discretionary Access Control, Introduction to Data Warehousing, OLAP, DataMining	10	14	0
		45	60	0

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- **Database Design Project:**
  Students design a database using ER diagrams, keys, and normalization.
- **SQL Query Exercise:**
  Students write and optimize SQL queries, including joins and aggregate functions.
- **Transaction Management Simulation:**
  Students simulate transactions and apply ACID properties and concurrency control.
- **NoSQL Exploration:**
  Students work with NoSQL databases (e.g., MongoDB) to perform CRUD operations.
- **Indexing and Performance Tuning:**
  Students analyze and implement indexing for optimizing database queries.

Suggested list of LAB Assignments (not limited to):

1. Draw an ER Diagram of Registrar Office
2. Draw an ER Diagram of Hospital Management System

3. Reduce The ER diagram in question no1 into tables
4. Reduce the ER diagram of question no2 into tables

Consider the following Schema

Supplier (SID, Sname, branch, city, phone)
Part(PID,Pname, color, price) Supplies(SID, PID, qty, date_supplied)

DDL Commands

5. Create the above tables
6. Add a new attribute state in supplier table
7. Remove attribute city from supplier table
8. Modify the datatype of phone attribute
9. Change the name of attribute city to address
10. Change a table's name, supplier to sup
11. Use truncate to delete the contents of supplies table
12. Remove the part table from database

DML Commands

1. Insert at least 10 records in tables supplier, part and supplies
2. Show the contents in tables supplier, part and supplies
3. Find the name and city of all suppliers
4. Find the name and phone no of all suppliers who stay in 'Delhi'
5. Find all distinct branches of suppliers
6. Delete the record of the supplier whose SID is 204001
7. Delete all records of supplier table
8. Delete all records of suppliers whose city starts with capital A.
9. Find the supplier names which have 'lk' in any position
10. Find the supplier name where 'R' is in the second position
11. Find the name of supplier whose name starts with 'V' and ends with 'A'
12. Change the city of all suppliers to 'BOMBAY'
13. Change the city of supplier 'Vandana' to 'Goa'

Queries with Constraints

1. Create the supplier table with Primary Key Constraint
2. Create supplies table with Foreign key Constraint
3. Create a part table with UNIQUE Constraint
4. Create supplier Table with Check Constraints
5. Create Supplier table with Default

Constraint Queries on TCL

1. Create Savepoints
2. Rollback to SavePoints

3. Use Commit to save on **Aggregate Functions:**

1. Find the minimum, maximum, average and sum of costs of parts
2. Count the total number of parts present
3. Retrieve the average cost of all parts supplied by 'Mike' **Queries**

on GROUP BY, HAVING AND ORDER BY Clauses

1. Display total price of parts of each color
2. Find the branch and the number of suppliers in that branch for branches which have more than 2 suppliers
3. Find all parts sorted by pname in ascending order and cost in descending order
4. Find the branch and the number of suppliers in that branch
5. Find out the 5th highest earning employee details.

Queries on Analytical, Hierarchical, Recursive nature.

1. Which department has the highest number of employees with a salary above $80,000, and what percentage of employees in that department have a salary above $80,000
2. Retrieve employee table details using the hierarchy query and display that hierarchy path starting from the top level indicating if it is a leaf and there exists a cycle.
3. What is the average salary for employees in the top 2 departments with the highest average salary, and what is the hierarchyof departments and sub-departments for these top 2 departments?
4. Use recursion to retrieve the employee table and display the result in breadth first and depth first order.
5. Write a recursive query to show the equivalent of level, connect_by_root and connect_by_path
6. Use recursion to retrieve the employee table and display the result in depth first order showing id, parent_id, level, root_id, path and leaf.

Queries on Operators

1. Find the pname, phoneno and cost of parts which have cost equal to or greater than 200 and less than or equal to 600.
2. Find the sname, SID and branch of suppliers who are in 'local' branch or 'global' branch
3. Find the pname, phone no and cost of parts for which cost is between 200 and 600
4. Find the pname and color of parts, which has the word'NET'anywhere in its pname.
5. Find the PID and pname of parts with pname either 'NUT' or 'BOLT'
6. List the suppliers who supplied partson '1st may 2000','12 JAN 2021', '17 dec 2000' ,'10 Jan 2021'
7. Find all the distinct costs of parts

Join Operators

1. Perform Inner join on two tables
2. Perform Natural Join on two tables
3. Perform Left Outer Join on tables
4. Perform Right Outer join on tables
5. Perform Full Outer Join on tables

Set Theory Operators

1. Show the use of UNION operator with union compatibility
2. Show the use of intersect operator with union compatibility
3. Show the use of minus operator with union compatibility
4. Find the cartesian product of two tables

Queries on Set Theory Operators

1. List all parts except 'NUT' and 'BOLT' in ascending order of costs
2. Display all parts that have not been supplied so far
3. To display the supplier names who have supplied 'green' part with cost 500 Rupees AND 'red' part with cost 400 Rupees.
4. To display the supplier names who have supplied 'green' part with cost 500 Rupees OR 'red' part with cost 400 Rupees.
5. To Display the name of suppliers who have supplied all parts that are 'red'incolor.

PL/SQL Programs

1. Write a PL/SQL Code to add two numbers
2. Write a PL/SQL code for Fibonacci series
3. Write a PL/SQL Code for greatest of 3 numbers
4. Write a PL/SQL code for area and circumference of acircle

PL/SQL Programs on Cursors

1. Write a Program using CURSOR todisplay SID and city of 1st record of supplier
2. Write a program using cursors to display the SID and City of all suppliers and then print the count of suppliers.

PL/SQL Programs on Cursors

1. Write a Program using CURSOR to display SID and city of 1st record of supplier
2. Write a program using cursors to display the SID and City of all suppliers and then print the count of suppliers.

PL/SQL Programs on Triggers, Procedures and Functions

1. Write a Program using TRIGGER on UPDATE
2. Write a command to See the effect of trigger
3. Write a Program using PROCEDURE to increase the cost by Rs.1000 for part whose PID is passed as an argument.
4. Write a procedure to update the city of an supplier whose SIDand city are passed as arguments and the procedure returns the name of supplier whose city is updated.
5. Write a function to return the total number of suppliers
6. Write a function to return the PID of part, for which the part name is passed
7. Write a function to find the sum total of costs of all parts.

PL/SQL Programs on Implicit Cursors

1. Insert a record using % ROWTYPE
2. Write a code using %NOTFOUND, %FOUND, %ROWCOUNT
3. Write a code using %TYPE

MongoDB Queries

1. Create a collection and insert documents into it using insertOne() and insertMany()
2. Select all documents in collection
3. Find the count of all suppliers
4. Find all records that have city='Delhi'
5. Retrieve all documents that have colore qual to 'red' or 'green'
6. Retrieve all documents where part_name is 'P1'or price is less than 200.
7. Update the record of 'Geeta', set city='Bombay' and phoneno='11223344'
8. Delete all records where price is greater than 5000
9. Display only the name and city of the supplier
10. Sort all suppliers on city and display only the first two records.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Describe database concepts, differentiate data models, and design ER diagrams and relational schemas with appropriate keys and constraints.

CLO2. Write SQL queries, apply relational algebra operations, and normalize database schemas up to BCNF using functional dependencies.

CLO3. Explain ACID properties, implement basic concurrency control methods, and choose suitable file organization and indexing strategies for performance tuning.

CLO4. Distinguish between relational and NoSQL databases, describe big data tools, and explain basic security and access control mechanisms in database systems.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO4				
Conceptual Knowledge			CLO2,CLO3			CLO1
Procedural Knowledge						
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	3	2	1	0	0	0	1	2	0
CLO2	3	2	2	3	0	1	0	2	1	0
CLO3	3	2	2	2	0	1	0	1	2	0
CLO4	3	2	2	2	2	1	0	2	2	0
Average Contribution	**3**	**2.25**	**2**	**2**	**0.5**	**0.75**	**0**	**1.5**	**1.75**	**0**

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Ramakrishnan, R., Gehrke, J., & Gehrke, J. (2003). *Database management systems* (Vol. 3). New York: McGraw-Hill.
2. Rosenzweig, B., & Silvestrova, E. (2004). *Oracle PL/SQL by example*. Prentice Hall

Professional.
3. Dayley, B. (2014). *NoSQL with MongoDB in 24 Hours, Sams Teach Yourself*. Sams publishing.

SEMESTER –III

CC-3-BCA-303: Software Engineering

Course Code : CC-3-BCA-303
Course Title : Software Engineering
Nature of the Course :CORE
Total Credit assigned : 3
Component of the Course : Lecture: 3 Practical: 0 Tutorial: 0

Course Objectives:

Objective-1. To understand the evolving role of software, different process models, and agile methodologies for software development.

Objective -2. To learn software requirements engineering, including functional and non-functional requirements, and the process of elicitation and validation.

Objective -3. To explore software design concepts, testing strategies, and quality management practices for creating high-quality software products.

Objective -4. To understand project planning, risk management, release management, and product sustenance in the software lifecycle.

Course Content:

Unit	Content	L	P	T
I	• The evolving role of software, changing nature of software, layered technology, a process framework, Process models: The waterfall model, incremental process models, evolutionary process models, spiral model. • **Agile software development**: Agility Principles, Agile methods, Plan-driven and agile development, Extreme programming, Scrum, A Tool Set for the Agile Process.	10	0	0
II	• **Software Requirements Engineering**: Functional and non-functional requirements, the software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management. • **Risk management**: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan. • **Project planning**- Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques.	11	0	0
III	• **Design:** Design process and design quality, design concepts, the design model, software architecture, data design,	13	0	0

	architectural design, Basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams. • **Testing Strategies**: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. • **Product metrics**: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.			
IV	• **Quality Management**: Quality concepts, software quality assurance, software reviews, formal Technical reviews, statistical software quality assurance, software reliability. • **Release Management**: Release planning, development and build plans, release strategies, risk management, and post-deployment monitoring. • **Product sustenance**: Maintenance, updates, End of life, migration strategies.	11	0	0
Total		45	0	0

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- **Process Model Comparison:**Compare and analyze different software development models (Waterfall, Agile, Incremental)
- **Requirements Elicitation:**Conduct a mock session to gather and validate software requirements.
- **Risk Management Exercise:**Identify risks and create a risk management plan (RMMM) for a software project.
- **Design Diagram Creation:**Create class, sequence, and use case diagrams for a software design.
- **Testing Strategy Development:**Develop test cases using black-box and white-box testing strategies.
- **Quality Assurance Review:**Perform a technical review focusing on software quality metrics.
- **Release Planning Simulation:**Simulate the planning and monitoring of a software release.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Apply various software development models, including agile and traditional models, to real-world projects.

CLO2. Demonstrate the ability to collect, analyze, and validate software requirements, ensuring the development of quality software.

CLO3. Design software architectures, create appropriate diagrams, and apply testing strategies for quality assurance.

CLO4. Effectively manage project risks, plan releases, and maintain software products through their lifecycle.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge			CLO1, CLO4	CLO2,		CLO3
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	3	3	0	0	2	3	0	0
CLO2	2	2	3	3	0	0	0	3	0	0
CLO3	2	3	3	3	0	0	0	3	0	0
CLO4	2	0	2	2	0	0	2	3	2	0
Average Contribution	2	1.75	2.75	2.75	0	0	1	3	0.5	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Software Engineering, N.S.Gill, Khanna Publishing House, 2023(AICTE Recommended Textbook)
2. Software Engineering, Ian Somerville, 9th edition, Pearson education.
3. Software Engineering A practitioner's Approach, 8th edition, Roger S Pressman, Bruce R.Maxim. Mc Graw Hill Education, 2015.

Reference Books

1. Stephen Schach, Software Engineering 7thed,McGraw-Hill,2007
2. Software Engineering: Principles and Practice Hansvan Vliet

SEMESTER –III

Course Code : SEC-4-BCA-301
Course Title : Python Programming
Nature of the Course : SEC
Total Credit assigned : 4
Component of the Course : Lecture: 2 Practical: 2 Tutorial: 0

Course Objectives:

Objective -1. Understand Python basics, strings, and control flow statements.
Objective-2. Learn functions, mutable/immutable objects, and use of Math and NumPy modules.
Objective -3. Handle file operations, exception handling, and perform data visualization

Course Content:

Unit	Content	L	P	T
I	• **Introduction**: History and Application areas of Python; Structure of Python Program; Identifiers and Keywords; Operators and Precedence; Basic Data Types and type conversion;Statements and expressions; Input/Output statements. • **Strings**: Creating and Storing Strings, Built-in functions for strings; string operators, String slicing and joining; Formatting Strings. • **Control Flow Statements:** Conditional Flow statements; Loop Control Statements; Nested control Flow; continue and break statements, continue, Pass and exit.	8	22	0
II	• **Functions:** Built-In Functions, Function Definition and call; Scope and Lifetime of Variables, Default Parameters, Command Line Arguments; Lambda Functions; Assert statement; Importing User defined module; • **Mutable and Immutable objects:** Lists, Tuples and Dictionaries; Commonly usedFunctions on Lists, Tuples and Dictionaries. Passing Lists, tuples and Dictionaries as arguments to functions. Using Math and Numpy module for list of integers and arrays.	12	18	0
III	• **Files:** Types of Files; Creating, Reading and writing on Text and Binary Files; The Pickle Module, Reading and Writing CSV Files. Reading and writing of csv and JSON files. • **Exception Handling:** Try-except-else-finally block, raise statement, hierarchy of exceptions, adding exceptions.Data visualization: Plotting various 2D and 3D graphics; Histogram; Pi charts; Sine and cosine curves.	10	20	0
Total		30	60	0

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- **String Manipulation Exercise:**Practice string creation, slicing, joining, and formatting using built-in string functions.
- **Control Flow Project:**Create a program that uses conditional and loop control statements, including nested flows, `continue`, and `break`.
- **Function Implementation:**Write functions with different parameter types (default, command-line, lambda), and use scope and lifetime concepts.
- **Data Structures Challenge:**Implement operations on lists, tuples, and dictionaries, passing them as arguments to functions.
- **File Handling Task:**Read from and write to text, binary, CSV, and JSON files, using the `Pickle` module for serialization.
- **Exception Handling Simulation:**Build a program with custom exceptions, using `try-except` blocks and the `raise` statement.
- **Data Visualization Project:**Create visualizations (histograms, pie charts, sine/cosine curves) using Python libraries like Matplotlib

Suggested list of LAB Assignments (not limited to):

1. Write a program to find whether a number is a prime number.
2. Write a program to print m raise to power n, where m and n are read from the user.
3. Write a program having a parameterised function that returns True or False depending on whether the parameter passed is even or odd.
4. Write a program to print the summation of the following series upto n terms:1-2+3-4+5-6+7...+n
5. Write a menu driven program to perform the following operations on strings using string built in functions.
   - Find the frequency of a character in a string.
   - Replace a character by another character in a string.
   - Remove the first occurrence of a character from a string.
   - Remove all occurrences of a character from a string.
6. Write a program that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string, then it should return -1
7. Using Numpy module write menu driven program to do following
   - Create an array filled with1's.
   - Find maximum and minimum values from an array
   - Dot product of 2 arrays.
   - Reshape a 1-D array to 2-D array.
8. Write a function that takes a sentence as input from the user and calculates the frequency of each letter. Use a variable of dictionary type to maintain the count.
9. Consider a tuplet1=(1,2,5,7,9,2,4,6,8,10).Write a program to perform followingoperations:
   - Print contents of t1 in 2 separate lines such that half values come on one line and other half in the next line.
   - Print all even values of t1 as another tuplet 2.

- Concatenate a tuplet 2= (11,13,15) witht1.
- Return maximum and minimum value from t1..

10. Write a function that reads a file file1 and copies only alternative lines to another file file2. Alternative lines copied should be the odd numbered lines.
11. Write a Python program to handle a Zero Division Error exception when dividing a number by zero.
12. Write a program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
13. Write a program that makes use of a function to display sine, cosine, polynomial and exponential curves.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Demonstrate proficiency in Python syntax, data types, control flow, and functions.
CLO2. Apply concepts of mutable and immutable objects, including lists, tuples, and dictionaries, and use libraries like Numpy for data manipulation.
CLO3. Manage files, handle exceptions, and create basic data visualizations using Python.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO1				
Conceptual Knowledge						
Procedural Knowledge			CLO2			CLO3
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	1	1	0	1	0	0	2	0
CLO2	2	0	1	1	0	0	0	0	0	0
CLO3	2	0	1	1	0	0	0	0	0	0
CLO4	2	0	2	2	0	0	0	0	0	0
Average Contribution	2	0.5	1.25	1.25	0	0.25	0	0	0.5	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books:

1. Venkatesh, NagarajuY,Introduction to Python Programming, Khanna Publishing House, 2021.
2. Jeeva Jose, Introduction to Computing & Problem Solving With PYTHON, Khanna Publishing House, 2023.
3. Sheetal Taneja & Naveen kumar: Python Programming a Modular approach–A Modular approach with Graphics, Database, Mobile and Web applications, Pearson, 2017.

Reference Books:

1. ThinkPython, by NAllen Downey N 2nd Edition,2015, O'Reilly. https://drive.google.com/file/d/1p9Pul6d5UvnQrO9-Q-LE2_p4YvMk5cIg/view
2. An introduction to Python for absolute beginners, by Bob Dowling, Cambridge Univ.

3. Introduction to Computation and Programming using Python, by John Guttag, 2nd edition, 2016, PHI India.

SEMESTER –III

Course Code : VAC-2-BCA-301
Course Title : Indian Constitution
Nature of the Course : VAC
Total Credit assigned : 2
Component of the Course : Lecture: 2 Practical: 0 Tutorial: 0

Course Objective:

Objective-1. To provide foundational knowledge of the making of the Indian Constitution, its structure, and the interpretation of its key features.
Objective-2. To understand the structure and functioning of the Union Government of India, including the roles of key constitutional functionaries.
Objective-3. To introduce the structure and operational dynamics of the state government and its administrative machinery.
Objective-4. To familiarize students with local governance systems and the structure of local administrative bodies.

Course Content:

Unit	Content	L	P
I	The Constitution-Introduction • The History of the Making of the Indian Constitution • Preamble and the Basic Structure, and its interpretation • Fundamental Rights and Duties and their interpretation	6	0
II	Union Government • Structure of the Indian Union • President–Role and Power • Prime Minister and Council of Ministers • Lok Sabha and Rajya Sabha	6	0
III	State Government • Governor–Role and Power • Chief Minister Power and Functions • Council of Ministers: composition, power and functions	5	0
IV	Local Administration • District Administration • Municipal Corporation • Zila Panchayat	7	0
V	Election Commission • Role and Functioning • Chief Election Commissioner • State Election Commission	6	0
		30	0

Mode of In-Semester assessment :

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

1. The Constitution – Introduction
   - Group Discussion: Importance of the Preamble in modern times.
   - Case Study Presentation: Landmark cases interpreting Fundamental Rights (e.g., Kesavananda Bharati case).
   - Quiz: Asymmetry and interpretation of Fundamental Duties and Directive Principles.
2. Union Government
   - Role Play: Simulation of a parliamentary session (Lok Sabha/Rajya Sabha debate).
   - Assignment: Write-up on the powers and responsibilities of the President vs. Prime Minister.
   - Class Presentation: Structure and functioning of the Indian Parliament.
3. State Government
   - Debate: Role of the Governor – ceremonial or executive?
   - Interview-Based Report: Students interact with local MLAs or Secretariat staff (virtual or real) and report their findings.
   - Flowchart Activity: Visual representation of state governance structure.
4. Local Administration
   - Field Visit / Virtual Tour: Visit to a local municipal office or panchayat office.
   - Poster Making: Responsibilities and functions of local administrative bodies.
   - Case Discussion: Examples of successful local governance models (e.g., Swachh Bharat urban initiatives).
5. Election Commission
   - Mock Election: Simulate an election process using EVMs or ballot papers.
   - Research Assignment: Profile of Chief Election Commissioners and their key reforms.
   - Infographic Creation: Structure and functions of the Election Commission and State Election Commissions.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain the historical background, understand the Preamble, interpret Fundamental Rights and Duties, and analyze the Directive Principles of State Policy.

CLO2. Describe the roles and powers of the President, Prime Minister, Council of Ministers, and the structure and functioning of Lok Sabha and Rajya Sabha.

CLO3. Identify the roles and responsibilities of the Governor, Chief Minister, Council of Ministers, and State Secretariat.

CLO4. Outline the structure, functions, and roles of District Administration, Municipal Corporations, and Zila Panchayats in local governance.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge	CLO3					
Conceptual Knowledge		CLO2,CLO4		CLO1		
Procedural Knowledge						
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	1	1	0	1	0	0	2	0
CLO2	2	0	1	1	0	0	0	0	0	0
CLO3	2	0	1	1	0	0	0	0	0	0
CLO4	2	0	2	2	0	0	0	0	0	0
Average Contribution	2	0.5	1.25	1.25	0	0.25	0	0	0.5	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Suggested Learning Resources:

1. Ethics and Politics of the Indian Constitution by Rajeev Bhargava,OxfordUniversityPress, New Delhi, 2008
2. Fadia, D. B. (2017). The constitution of India. *Sahitya Bhavan publication Agra*.
3. Basu, D. D., Manohar, V. R., Banerjee, B. P., & Khan, S. A. (2015). *Introduction to the Constitution of India* (Vol. 163). Gurgaon: LexisNexis.

Suggested Software/Learning Websites:

a. https://www.constitution.org/cons/india/const.html
b. http://www.legislative.gov.in/constitution-of-india
c. https://www.sci.gov.in/constitution
d. https://www.toppr.com/guides/civics/the-indian-constitution/the-constitution-of-india/

Cases

- Rustom Cavasjee Cooper v.Union of India, (1970) 1 SCC 248
- State of Rajasthan v. Mohan Lal Vyas, AIR 1971 SC 2068 (confirmation of a private monopoly, not a violation of fundamental right)
- Mithilesh Garg v. Union of India, (1992) 1 SCC 168: AIR 1992 SC 221 (Right to carry on business, not breached when it is liberalised)
- Chintamanrao v. The State of Madhya Pradesh, AIR 1951 SC 118 (scope of reasonable restrictions in relation to trade and occupation)
- Cooverjee B. Bharucha v. Excise Commissioner, Ajmer, AIR 1954 SC 220 (the reasonableness of the restriction imposed may depend upon the nature of the business and prevailing conditions including public health and morality)
- T. B. Ibrahim v. Regional Transport Authority. Tanjore, AIR 1953 SC 79
- Harman Singh v. RTA, Calcutta, AIR 1954 SC 190
- Dwarka Prasad Laxmi Narain v. State of U.P., AIR 1954 SC 224
- State of Bombay v. R.M.D. Chamarbaugwala, AIR 1957 SC 699

- Parbhani Transport Coop. Society Ltd. v. Regional Transport Authority, Aurangabad, AIR 1960 SC 801
- State of Bombay v. R. M. D. Chamarbaugwala, (1957) S.C.R. 874,
- G.K.Krishnan vs State of Tamil Nadu, 1975 SCC(1) 375
- Automobile Transport (Rajasthan) Ltd. Vs State of Rajasthan, AIR 1962 SC 1406

SEMESTER–IV

CC-5-BCA-401: Computer Networks

Course Code : **CC-5-BCA-401**
Course Title : **Computer Networks**
Nature of the Course : **CORE**
Total Credit assigned : **5**
Component of the Course : **Lecture: 3 Practical: 2 Tutorial: 0**

Course Objectives:

Objective-1. To introduce the basic concepts, architecture, models, topologies, transmission modes, and networking devices in computer networks.
Objective -2. To provide foundational understanding of data link layer operations, Ethernet standards, addressing schemes, and network protocols including ARP and VLANs.
Objective -3. To explore IP routing, transport protocols, congestion control mechanisms, QoS, and fundamental network security practices.

Objective -4. To examine application layer protocols, real-world network applications, emerging networking technologies, and network management tools.

Course Content:

Unit	Content	L	P	T
I	**Introduction to Computer Networks** • **Overview of Computer Networks:** Definition and Objectives, Applications and Examples Network Components and Architecture • **Network Models:** OSI Model: Layers and Functions, TCP /IP Model: Layers and Functions Comparison between OSI and TCP/IP Models • **Network Topologies:** Physical vs. Logical Topologies, Common Topologies: Star, Ring, Bus, Mesh, Hybrid, Advantagesand Disadvantages of Each Topology • **Data Transmission:** Analog vs. Digital Signals, Transmission Modes: Simplex, Half-Duplex, Full-Duplex, Bandwidth and Latency • **Networking Devices:** Routers, Switches, Hubs, Bridges, Gateways, Functions and Configurations of Each Device.	10	16	0
II	**Data Link Layer and Networking Protocols** • **Data Link Layer Fundamentals:** Functions of the Data Link Layer, Framing, Error Detection, and Error Correction, Flow Control Mechanisms. • **Ethernet:**Ethernet Standards and Frame Structure, MAC Addressing and ARP, Ethernet Switching: Basic Concepts and Methods • **Network Protocols:** Introduction to TCP/IP Protocol Suite, IP	10	14	0

	Addressing: IPv4 and IPv6 Subnetting and CIDR Notation • **Address Resolution Protocol (ARP):** ARP Operation and Table, ARP Spoofing and Security Considerations • **Virtual LANs (VLANs):** Concept of VLANs, VLAN Taging and Configuration, Benefits and Use Cases			
III	**Network Layer and Transport Layer** • **Network Layer:** IP Routing: Static vs.Dynamic Routing, Routing Protocols: RIP, OSPF, BGP, Network Address Translation (NAT) • **Transport Layer:** TCP vs.UDP: Characteristics and Use Cases, TCP Handshake and Connection Management, Flow Control and Congestion Control in TCP • **Congestion Control Algorithms:** Techniques: Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery, TCP Variants: TCP Reno, TCP Vegas. • **Quality of Service (QoS):** QoS Principles and Mechanisms, Differentiated Services (DiffServ) and Integrated Services (IntServ) • **Network Security Fundamentals:** Threats and Vulnerabilities, Basic Security Mechanisms: Firewalls, VPNs, Encryption	12	16	0
IV	**Application Layer and Emerging Technologies** • **Application Layer Protocols:** HTTP/HTTPS: Structure and Operation, FTP, SMTP, POP3, IMAP: Protocols and Uses, DNS: Domain Name System and Resolution • **Network Applications:** Web Browsing, Email Communication, File Transfer, Voice over IP (VoIP) and Streaming. • **EmergingTechnologies:**Software-DefinedNetworking(SDN),NetworkFunction Virtualization (NFV), Internet of Things (IoT) and Its Impact on Networking • **Network Management:** SNMP: Simple Network Management Protocol, Network Monitoring Tools and Techniques. • **Future Trends in Networking:** 5G and Beyond, Network Automation and Artificial Intelligence in Networking	13	14	0
Total		45	60	0

Mode of In-Semester assessment :

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

1. **Network Design Exercise:**Design a network using various topologies (e.g., star, mesh) and select appropriate devices (router, switch, hub).
2. **Protocol Configuration Task:**Implement and configure basic protocols like TCP/IP, ARP, and VLANs on a simulated network.
3. **Troubleshooting Exercise:**Troubleshoot a given network setup, identifying issues with IP addressing, routing, and device configuration.

4. **Packet Sniffing Lab:**Use network monitoring tools (e.g., Wireshark) to capture and analyze network traffic for different protocols (HTTP, FTP, etc.).
5. **Routing Protocol Simulation:**Simulate routing protocols (RIP, OSPF) and demonstrate dynamic IP routing in a small network.
6. **Network Security Assessment:**Set up basic security measures like firewalls and VPNs, and assess network vulnerabilities.

Suggested list of LAB Assignments (not limited to):

1. Configure Basic Network Settings:
   a) IP Address Configuration
   b) Subnet Mask and Gateway Settings
2. Implement Network Protocols:
   a) Write a simple Python script to perform DNS resolution.
   b) Implement a basic HTTP client-server application.
3. Network Simulation:
   a) Use network simulation tools (e.g.,Cisco Packet Tracer) to design and simulate network topologies.
   b) Configure routers and switches in a simulated environment.
4. Performance Measurement:
   a) Measure network performance using tools like`ping`, `traceroute`, and `iperf`.
   b) Analyze network traffic using Wireshark.
5. Implement VLANs:
   a) Configure VLANs on a switch and verify using simulation tools.
6. SetUp a Simple Web Server:
   a) Deploy a basic web server and configure HTTP/HTTPS access.
7. Network Security Lab:
   a) Implement basicfirewall rules and VPN configurations.
   b) Perform vulnerability scanning and analyze results.
8. Network Troubleshooting:
   a) Diagnose and resolve common network issues.
   b) Use troubleshooting commands and techniques to fix connectivity problems

Expected Course Learning Outcomes:
By the end of the course, students will be able to:

CLO1. Explain the structure and functioning of computer networks, compare OSI and TCP/IP models, identify network topologies, and describe the roles of common networking devices.
CLO2. Analyze data link layer functions, implement subnetting, understand MAC addressing, and configure basic VLANs and ARP settings.
CLO3. Differentiate between routing protocols and transport layer protocols, explain congestion control strategies, and apply basic network security concepts such as firewalls and VPNs.

CLO4. Describe the operation of application layer protocols, evaluate the role of technologies like SDN, IoT, and 5G, and apply tools for monitoring and managing network performance.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						

Conceptual Knowledge			CLO3	CLO1	CLO4	
Procedural Knowledge				CLO2		
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	1	0	0	1	0	0	2	0
CLO2	2	2	2	0	0	0	0	0	1	0
CLO3	2	0	0	0	2	1	0	0	2	0
CLO4	2	2	1	2	0	0	0	0	1	0
Average Contribution	2	1.5	1	0.5	0.5	0.5	0	0	1.5	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books:

1. Tanenbaum, A. S. Computer Networks 5th ed.(2011)..
2. Kurose, J. F., & Ross, K. W. (2007). Computer networking: A top-down approach edition. *Addision Wesley*, *12*..

Reference Books:

1. Forouzan, B. A. (2007). *Data communications and networking*. HugaMedia..
2. Peterson, L. L., & Davie, B. S. (2007). *Computer networks: a systems approach*. Elsevier.

SEMESTER–IV

Course Code : **CC-3-BCA-402**
Course Title : **Design and Analysis of Algorithms**
Nature of the Course : **CORE**
Total Credit assigned : **3**
Component of the Course : **Lecture: 3 Practical: 0 Tutorial: 0**

Course Objectives:

Objective -1. To introduce the foundational concepts of algorithm design and performance evaluation using time and space complexity, and asymptotic notations.
Objective -2. To study the principles of divide & conquer and greedy algorithm design paradigms, along with their real-world applications and performance analysis.
Objective -3. To understand the dynamic programming approach and explore fundamental graph algorithms for traversal and optimization problems.
Objective -4. To explore algorithmic problem-solving using backtracking and understand the theoretical limitations of algorithms through complexity classes.

Course Content:

Unit	Content	L	P	T
I	Introduction: • What is an algorithm? Design and performance analysis of algorithms, time complexity, space complexity. • Asymptotic notations (O, Ω, Θ) to measure growth of a function and application to measure complexity of algorithms. • Analysis of sequential search, bubble sort, selection sort, insertion sort, matrix multiplication. Recursion: Basic concept. Analysis of recursive algorithms, Master's theorem	11	0	0
II	The Divide & Conquer Design Technique: • The general concept. Binary search, finding the maximum and minimum, merge sort, quick sort. Best and worst case analysis for the mentioned algorithms. Strassen's matrix multiplication. Lower bound for comparison-based sorting. The Greedy Design Technique: • The general concept. Applications to general Knapsack problem, finding minimum weight spanning trees: Prim's and Kruskal's algorithms, Dijkstra's algorithm for finding single source shortest paths problem	12	0	0

III	The Dynamic Programming Design Technique: • The general concept. Computation of Fibonacci series and Binomial coefficients, all pair shortest paths problem (Floyd-Warshall's algorithm), 0/1 Knapsack problem. Algorithms on Graphs: • Breadth First Search, Depth First Search, finding connected components, DFS of a directed graph, topological sorting.	11	0	0
IV	**Limitations of Algorithmic Power:** • Backtracking Method: n-Queen problem; sum of subsets problem/Hamiltonian circuit • Problem /vertex cover problem. • Computational Intractability: Overview of non-deterministic algorithms, P, NP, NP-Complete and NP-hard problems.	11	0	0
		45	0	0

Mode of In-Semester assessment :

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

1. **Algorithm Complexity Calculation:**Analyze the time and space complexity of basic algorithms using asymptotic notations.
2. **Sorting Algorithm Implementation:**Implement and test bubble sort, insertion sort, and measure their performance.
3. **Divide and Conquer Exercise:**Implement merge sort and quick sort, then compare their efficiency.
4. **Greedy Algorithm Application:**Solve the knapsack problem using a greedy approach and analyze its efficiency.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Analyze basic algorithms using asymptotic notations and apply complexity analysis techniques to recursive and iterative algorithms.

CLO2. Implement and evaluate divide & conquer and greedy algorithms such as merge sort, quick sort, Kruskal's, and Dijkstra's algorithms.

CLO3. Solve optimization problems using dynamic programming and apply graph traversal techniques like BFS and DFS for various applications.

CLO4. Implement backtracking algorithms and classify problems based on computational complexity into P, NP, NP-Complete, and NP-Hard categories.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO3				
Conceptual Knowledge		CLO4				
Procedural Knowledge						CLO1, CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	3	2	0	0	0	0	0	0	0
CLO2	3	3	2	0	0	0	0	0	0	0
CLO3	3	3	3	3	0	0	0	0	0	0
CLO4	3	3	2	0	0	0	0	0	0	0
Average Contribution	3	3	2.25	0.75	0	0	0	0	0	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books

1. Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (2009). Stein, Clifford Introduction to Algorithms.
2. Levitin, A. (2008). *Introduction to design and analysis of algorithms, 2/E*. Pearson Education India.

Reference Books

1. Kleinberg, J., & Tardos, E. (2006). *Algorithm design*. Pearson Education India.

SEMESTER–IV

Course Code : CC-5-BCA-403
Course Title : Artificial Intelligence
Nature of the Course : CORE
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives:

Objective -1. Understand the fundamentals of AI, including intelligent agents and problem-
Objective -2. Learn advanced search algorithms and optimization techniques like A*, Minimax, and genetic algorithms.

Objective -3. Gain knowledge of logical reasoning, uncertainty handling, and AI-based planning systems.

Objective -4. Explore domains of AI such as machine learning, computer vision, robotics, and expert systems, with an understanding of legal and ethical issues.

Course Content:

Unit	Content	L	P	T
I	Introduction to Al • What is AI? Intelligent Agents: Agents and environment, the concept of Rationality, the nature of environment, the structure of Agents. Knowledge-Based Agents**:** Introduction to Knowledge-Based Agents,The Wumpus World as an Example World. Problem-solving: Problem-solving agents	11	12	0
II	Advanced Search Techniques • Uninformed Search: DFS, BFS, Iterative Deepening Search. Informed Search: Best First Search, A* search, AO* search. Adversarial Search & Games: Two-player zero-sum games, Minimax Search, Alpha-Beta pruning.Constraints and Constraint Satisfaction Problems (CSPs), Backtracking search for CSP. Evolutionary Search Techniques**:** Introduction to evolutionary algorithms, Genetic algorithms, Applications of evolutionary search in AI.	12	18	0
III	Logical Reasoning and Uncertainty • Logic: Propositional logic, First-order predicate logic, Propositional versus first-order inference, Unification and lifting. Inference: Forward chaining, Backward chaining, Resolution, Truth maintenance systems.Introduction to Planning: Blocks World problem, Strips; Handling Uncertainties: Non-monotonic reasoning, Probabilistic reasoning, Introduction to Fuzzy set theory	11	16	0

IV	Domains and Applications of AI • Domains in AI: Introduction to Machine Learning, Computer Vision, Robotics, Natural Language Processing, Deep Neural Networks, and their Applications. Expert Systems: The architecture and role of expert systems include two case studies. Legal and Ethical Issues: Concerns related to AI	11	14	0
		45	60	0

Mode of In-Semester assessment :

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

1. **Search Algorithm Implementation:**Implement DFS, BFS, and A* search algorithms and solve simple path finding problems.
2. **Minimax Algorithm with Alpha-Beta Pruning:**Simulate a two-player game (e.g., Tic-Tac-Toe) using Minimax search and optimize it with Alpha-Beta pruning.
3. **CSP Problem Solving:**Solve a Constraint Satisfaction Problem (CSP) using backtracking search (e.g., Sudoku or map coloring).
4. **Genetic Algorithm Application:**Implement a basic genetic algorithm to solve an optimization problem (e.g., finding the best-fit solution to a traveling salesman problem).
5. **Logical Reasoning Exercise:**Use forward and backward chaining to infer facts in a knowledge-based agent (e.g., BlocksWorld problem).
6. **Probabilistic Reasoning Simulation:**Implement a simple probabilistic reasoning task (e.g., Bayesian network for disease diagnosis).

Suggested list of LAB Assignments (not limited to):

Prerequisites: Basic understanding of algorithms and data structures (e.g., trees, graphs, lists). Proficiency in Python programming, including libraries like NLTK for NLP tasks.

LAB Experiments

The lab experiments may be implemented in Python. Libraries like NLTK, Tensorflow and Keras may be used for Machine learning experiments.

1. Demonstrate basic problem-solving using Breadth-First Search on a simple grid.
2. Implement Depth-First Search (DFS) on a small graph.
3. Solve the Water Jug Problem using Breadth First Search (BFS).
4. Implement a Hill Climbing search to find the peak in a numeric dataset.
5. Apply the A* Search algorithm to find the shortest path in a 4x4 grid.
6. Implement the Minimax search algorithm for 2-player games. You may use a game tree with 3 plies.
7. Solve the 4–Queens Problem as a CSP backtracking problem.
8. Use constraint propagation to solve a Magic Square puzzle.
9. Apply optimization techniques to find the maximum value in a list.
10. Represent and evaluate propositional logic expressions.
11. Implement a basic rule-based expert system for weather classification.
12. Implement a basic AI agent with simple decision-making rules.
13. Implement a basic Rule-Based Chatbot.

14. Using Python NLTK, perform the following Natural Language Processing (NLP) tasks for text content.
    a) Tokenizing
    b) Filtering Stop Words
    c) Stemming
    d) Part of Speech tagging
    e) Chunking
    f) Named Entity Recognition(NER)
15. Perform Image classification for a given dataset using CNN. You may use Tensorflow /Keras.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Identify and explain the concepts of AI, intelligent agents, and their environments.

CLO2. Implement and evaluate various search algorithms like DFS, BFS, A*, and genetic algorithms.

CLO3. Apply logical reasoning techniques and handle uncertainties using probabilistic and fuzzy set theory.

CLO4. Understand and apply AI techniques in real-world domains such as NLP, robotics, and expert systems.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge	CLO1	CLO1, CLO4				
Procedural Knowledge			CLO3, CLO4		CLO2	CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	0	0	2	1	0	0	2	3
CLO2	3	2	3	0	0	0	0	0	0	0
CLO3	2	0	0	0	0	0	0	0	2	0
CLO4	2	0	0	0	0	0	0	0	2	0
Average Contribution	2.25	0.5	0.75	0	0.5	0.25	0	0	1.5	0.75

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

Text Books:

1. Nilsson, N. J. (1998). *Artificial intelligence: a new synthesis*. Elsevier.
2. Patterson, D. (1990). *Introduction to artificial intelligence and expert systems*. Prentice-

Hall, Inc..

SEMESTER–IV

Course Code : **SEC-2-BCA-401**
Course Title : **Design Thinking and Innovation**
Nature of the Course : **SEC**
Total Credit assigned : **2**
Component of the Course : **Lecture: 1 Practical: 0 Tutorial: 1**

Course Objectives:

Objective -1. To introduce the fundamentals of Design Thinking, its importance in innovation, customer-centricity, and successful real-world applications.
Objective -2. To help students understand the role of empathy in identifying and defining complex real-world problems through observation and individual analysis.
Objective -3. To develop the ability to generate creative ideas, build rapid prototypes, and plan implementation using Design Thinking strategies.
Objective -4. To emphasize iterative design by incorporating user feedback and focusing on user experience, ergonomic design, and final concept presentation.

Course Content:

Unit	Content	L	P	T
I	Basics of Design Thinking • Understand the concept of innovation and its significance in business • Understanding creative thinking process and problem solvingapproaches • Know Design Thinking approach and its objective • Design Thinking and customer centricity – real world examples of customer challenges, use of Design Thinking to Enhance Customer Experience, Parameters of Product experience, Alignment of Customer Expectations with Product. • Discussion of a few global success stories like AirBnB, Apple, IDEO, Netflix etc. • Explain the four stages of Design Thinking Process – Empathize, Define, Ideate, Prototype, Implement	4	0	4
II	Learning to Empathize and Define the Problem • Know the importance of empathy in innovation process – how can students develop empathy using design tools • Observing and assimilating information • Individual differences & Uniqueness Group Discussion and Activities to encourage the understanding, acceptance and appreciation of individual differences. • What are wicked problems • Identifying wicked problems around us and the potential impact of their solutions	4	0	4
III	Ideate, Prototype and Implement • Know the various templates of ideation like brainstorming,	3	0	3

	systems thinking • Concept of brainstorming– how to reach consensus on wicked problems • Mapping customer experience for ideation • Know the methods of prototyping, purpose of rapid prototyping. • Implementation			
IV	Feedback, Re-Design & Re-Create • Feedback loop, focus on User Experience, address ergonomic challenges, user focused design • Final concept testing, • Final Presentation – Solving Problems through innovative design concepts & creative solution	4	0	4
		15	0	15

Mode of In-Semester assessment :

A. Sessional Activity

B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

A. Basics of Design Thinking

1. **Case Study Discussion**
   - Analyze innovation stories (AirBnB, Apple, Netflix) to understand Design Thinking in action.
2. **Innovation Brainstorming**
   - Group activity to ideate solutions for user needs using creative thinking methods.

B. Empathize and Define the Problem

3. **Empathy Mapping**
   - Observe users and create empathy maps to understand their challenges.
4. **Wicked Problem Identification**
   - Identify and discuss complex real-life problems with high impact potential.
5. **Understanding Differences**
   - Group discussion or role-play to appreciate diverse perspectives.

C. Ideate, Prototype and Implement

6. **Ideation Workshop**
   - Use brainstorming and mind mapping to generate innovative ideas.
7. **Customer Journey Mapping**
   - Visualize user experience to refine ideas.
8. **Rapid Prototyping**
   - Build quick, low-cost models of solutions for peer feedback.

D. Feedback and Redesign

9. **User Testing**
   - Test prototypes with users and gather feedback for improvement.
10. **Final Presentation**
    - Present complete design solutions covering all stages of the Design Thinking process.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain the core concepts of Design Thinking and illustrate its application using global success stories and customer-centered product development.
CLO2. Apply empathy-based tools to observe, understand, and define "wicked problems" considering diverse user perspectives.
CLO3. Create innovative solutions by using ideation techniques, prototyping tools, and implementation strategies for solving identified problems.
CLO4. Refine and present user-centric design solutions through feedback loops, ergonomic analysis, and final presentations.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge		CLO1				
Procedural Knowledge			CLO2		CLO4	CLO3, CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	3	2	0	0	1	0	0	2	0
CLO2	0	3	3	3	0	0	0	0	1	0
CLO3	0	3	3	3	0	0	0	0	1	0
CLO4	0	3	3	0	0	2	0	0	0	0
Average Contribution	0.5	3	2.75	1.5	0	0.75	0	0	1	0

(3 = High, 2 = Moderate, 1 = Low, 0 = No direct contribution)

TextBooks:

1. Balagurusamy, E. (2023). Developing thinking skills: The way to success. Khanna Book Publishing Company.
2. Brown, T. (2009). Change by design: How design thinking creates new alternatives for business and society. Harvard Business Press.
3. Krishnan, R. T., & Dabholkar, V. (2013). 8 steps to innovation: Going from jugaad to excellence. Collins Business.

Semester-V

SEC-2-BCA-501: Quantitative Techniques

Course Code : SEC-2-BCA-501
Course Title : Quantitative Techniques
Nature of the Course : SEC
Total Credit assigned : 2
Component of the Course : Lecture: 0 Practical: 0 Tutorial: 2

Course Objectives

The main objectives of this course are:

Objective -1. Introduce quantitative models for decision making in business, computing and project contexts.

Objective -2. Develop ability to formulate and solve linear programming, transportation and assignment problems.

Objective -3. Enable students to use network, decision and simple simulation techniques for structured problem solving.

Course Content

Unit	Content	L	T	P
I	**Introduction to Quantitative Techniques and Model Building** • Meaning, scope and role of quantitative techniques in decision making; data, variables, constraints, objective functions and assumptions; steps in quantitative problem solving: problem definition, model formulation, solution, validation and interpretation; applications of quantitative techniques in computing, education, business, logistics and project management.	3	3	0
II	**Linear Programming and Optimization** • Introduction to linear programming; formulation of simple linear programming problems; objective function and constraints; feasible region and optimal solution; graphical solution method for two-variable problems; interpretation of results. Basic concept of simplex method and simple tabular form without detailed computational complexity.	5	5	0
III	**Transportation, Assignment and Network Techniques** • Transportation model: meaning, structure and formulation; initial feasible solution using simple methods such as North-West Corner Method and Least Cost Method. Assignment problem: meaning, structure and basic solution approach. Introduction to network models; concept of shortest path and maximum flow with simple applications. PGantt ERT and CPM networks; critical path, project duration and basic concept of floats. Applications of project scheduling in software and IT projects.	7	7	0
	Total:	15	15	0

Note: *Topics should be taught at an introductory and application-oriented level. Advanced numerical problems, detailed simplex computations, complex transportation and assignment cases, and detailed maximum flow algorithms may be avoided.*

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests / Viva-voce / Presentation, as applicable

Sessional activity:

1. Solve tutorial sheets on LP formulation, graphical solution and interpretation of constraints.
2. Prepare spreadsheet-based solutions for transportation, assignment and decision-tree problems.
3. Analyze a small project network and identify critical path and floats.
4. Present a quantitative decision case using tables, charts and interpretation.

Evaluation Criteria:

- Correct model formulation and assumptions.
- Accuracy of solution method and calculations.
- Ability to interpret results in a decision-making context.
- Neat presentation of tutorial work and case reports.

Tutorial Work:

1. Formulate five real-life decision problems as objective functions and constraints.
2. Solve at least two LPPs using graphical method and spreadsheet Solver.
3. Solve transportation and assignment problems using tabular methods.
4. Prepare a PERT/CPM network for a small software project.
5. Create a decision tree for an academic or business decision case.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain the role of quantitative techniques in structured decision making.

CLO2. Formulate and solve basic linear programming models.

CLO3. Apply transportation, assignment and network techniques to practical problems.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge		CLO1				
Procedural Knowledge			CLO2, CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	0	0	0	1	0	0	2	0
CLO2	3	3	2	2	0	0	0	0	0	0
CLO3	3	2	0	0	0	0	0	0	0	0
Average Contribution	3	2.33	0.67	0.67	0	0.33	0	0	0.67	0

(3=High, 2= Moderate, 1= Low, -=No direct correlation)

Text Books:

1. Taha, H. A. (2017). Operations research: An introduction (10th ed.). Pearson.
2. Hillier, F. S., & Lieberman, G. J. (2021). Introduction to operations research (11th ed.). McGraw-Hill.
3. Ragsdale, C. T. (2015). Spreadsheet modeling and decision analysis: A practical introduction to business analytics. Cengage Learning.

Reference Books:

1. Winston, W. L. (2004). Operations research: Applications and algorithms (4th ed.). Cengage Learning.
2. Sharma, J. K. (2017). Operations research: Theory and applications (6th ed.). Laxmi Publications.
3. Vohra, N. D. (2021). Quantitative techniques in management (6th ed.). McGraw-Hill Education.

Semester-V

SEC-4-BCA-502: Internship/Capstone Project

Course Code	**: SEC-4-BCA-502**
Course Title	**: Internship/Capstone Project**
Nature of the Course	**: SEC**
Total Credit assigned	**: 4**
Component of the Course	**: Lecture: 0 Practical: 4 Tutorial: 0**

Course Objectives

The main objectives of this course are:

Objective -1. Provide industry, research, academic or community-based exposure through an eight-week internship/capstone experience.

Objective -2. Enable students to apply computing knowledge to a real-world problem or organizational task.

Objective -3. Develop professional skills such as documentation, teamwork, communication, ethics and time management.

Objective -4. Prepare students to produce an internship/capstone report and present their work through viva-voce or demonstration.

Course Content

Internship/Capstone Work:

• The student shall complete the prescribed internship/capstone work under the supervision of a faculty guide and, wherever applicable, an industry/research/community mentor.

• The work may be carried out in an industry, research organization, academic laboratory, start-up, NGO/community organization, government/ public digital-service environment, or through an approved virtual/remote internship mode.

• The work shall involve application of computer-application knowledge to a real problem, professional task, software/data activity, technical documentation, digital-service support, or approved capstone deliverable.

• The student shall maintain a logbook/activity diary, obtain mentor verification where applicable, and submit a structured internship/capstone report with evidence of work.

• Evaluation shall be based on regularity, mentor/guide feedback, logbook, report, presentation, demonstration/viva-voce, and quality of learning outcomes, as prescribed by the University/Department.

Sessional activity:

1. Submission of internship/capstone proposal and work plan.
2. Weekly progress logbook and mentor interaction.
3. Interim presentation on problem, methodology and progress.
4. Final report submission, demonstration and viva-voce.

Evaluation Criteria:

- Regularity, discipline and professional conduct during the internship/capstone work.
- Quality of problem understanding, planning and implementation.
- Completeness of logbook, report, artifacts and presentation.
- Ability to explain work, learning, limitations and future scope.

Internship/Capstone Work:

1. Complete an eight-week internship/capstone in industry, research institution, academic laboratory, NGO/community organization or approved virtual mode.
2. Maintain a daily/weekly activity diary with mentor verification.
3. Develop a software application, data-analysis output, research study, community digital solution, technical documentation or equivalent approved deliverable.
4. Submit internship/capstone report with certificates, problem description, methodology, implementation, results and reflection.
5. Present final work before the departmental evaluation committee.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Identify a real-world computing problem or professional task and prepare a feasible work plan.
CLO2. Apply computing tools, programming, data handling or documentation skills in an internship/capstone setting.
CLO3. Demonstrate professional conduct, teamwork, communication and ethical responsibility.
CLO4. Prepare and present a structured internship/capstone report with appropriate evidence of work.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge				CLO1		
Procedural Knowledge			CLO2,CLO3		CLO4	CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	0	0	1	0	0	0	1	2	0	0
CLO2	2	2	2	3	0	2	2	3	1	0
CLO3	0	0	2	0	2	2	2	3	0	0
CLO4	0	0	2	0	0	2	2	3	0	0
Average Contribution	**0.5**	**0.5**	**1.75**	**0.75**	**0.5**	**1.5**	**1.75**	**2.75**	**0.25**	**0**

(3=High, 2= Moderate, 1= Low, -=No direct correlation)

Semester-V

SEC-4-BCA-601: Major Project - Phase I: Initiation and Design

Course Code	**: SEC-4-BCA-601**
Course Title	**: Major Project - Phase I: Initiation and Design**
Nature of the Course	**: SEC**
Total Credit assigned	**: 0 (Initiation phase; evaluation carried in Semester VI)**
Component of the Course	**: Lecture: - Practical: 2 Tutorial: -**

Course Objectives

The main objectives of this course are:

Objective -1. Guide students to identify an appropriate major project problem in computing or applications of computing.
Objective -2. Develop ability to prepare a proposal, requirement specification and project design.
Objective -3. Encourage disciplined use of software engineering, research methodology, documentation and ethical practices.
Objective -4. Prepare students for implementation and evaluation in the sixth semester.

Course Content

Project Work - Phase I:

- The fifth-semester major project component shall be treated as the initiation, proposal, requirement-analysis and design phase of the project to be evaluated in the sixth semester.
- Students shall select a feasible computing/application problem under a faculty guide, either individually or in an approved group, as per departmental norms.
- The work shall include problem identification, objective formulation, literature/background study, scope definition, requirement analysis, methodology, tool selection and project planning.
- Students shall prepare project synopsis, requirement specification, design documents, expected deliverables, timeline and ethical/data-use considerations.
- Prepare the Software Requirements Specification (SRS), Entity-Relationship (ER) Diagram, and Data Flow Diagram (DFD), or any other relevant design and documentation artifacts, as recommended by the project supervisor and required by the nature of the proposed project.
- Departmental review shall be conducted through proposal presentation, guide feedback and approval of the project plan before implementation in the sixth semester.

Text Books:

1. Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
2. Sommerville, I. (2016). Software engineering (10th ed.). Pearson.

Semester-VI

CC-4-BCA-601: Generative AI: Generative AI

Course Code : CC-4-BCA-601
Course Title : Generative AI
Nature of the Course : Core Course
Total Credit assigned : 4
Component of the Course : Lecture: 2 Practical: 2 Tutorial: 0

Course Objectives

The main objectives of this course are:

Objective -1. Introduce the concepts, models and application areas of generative artificial intelligence.
Objective -2. Develop understanding of neural-network based generation, attention mechanisms, transformers and large language models.
Objective -3. Enable students to apply prompt engineering, retrieval-augmented generation and basic generative tools responsibly.
Objective -4. Develop hands-on ability to build small generative AI applications with ethical, privacy and security awareness.

Course Content

Unit	Content	L	T	P
I	Foundations of Generative AI • AI, machine learning, deep learning and generative AI • Generative vs. discriminative models • Foundation models, pre-training, fine-tuning and inference • Tokens, embeddings and vector representations • Applications: text generation, summarization, coding assistance, image generation, education, business and creativity • Risks: hallucination, bias, privacy, copyright and misuse	8	0	8
II	Neural Architectures for Generation • Activation functions gradient descent • Review of neural networks, CNN, RNN, LSTM and sequence learning • Attention mechanism and transformer architecture • Autoencoders and variational autoencoders: basic idea • Generative adversarial networks: generator, discriminator, loss and training issues • Diffusion model concept and comparison of generative approaches • Evaluation and limitations of generated output	8	0	16
III	Large Language Models, Prompt Engineering and RAG • Large language models and conversational AI • Prompt design: instruction, context, examples, constraints and output format • Zero-shot, one-shot and few-shot prompting • Responsible use of stepwise prompts without exposing private reasoning • Embeddings, semantic search and vector databases • Retrieval-Augmented Generation: indexing, retrieval, grounding and	8	0	16

	response generation • Basic evaluation of LLM responses			
IV	Generative AI Applications and Responsible Deployment • Building simple GenAI applications: chatbot, summarizer, content assistant and code assistant • Multimodal generation: text-to-image and image understanding overview • Use of open-source models and cloud APIs • Guardrails, content safety, prompt injection and data leakage risks • Bias, fairness, explainability, transparency and academic integrity • Mini-project design, testing, report and demonstration	6	0	20

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests / Viva-voce / Presentation, as applicable

Sessional activity:

1. Prompt design and evaluation assignment.
2. Hands-on exercises using an open-source or cloud-based generative AI tool.
3. Mini-project proposal on a generative AI application.
4. Viva-voce and demonstration of generative AI use cases and risks.

Evaluation Criteria:

- Conceptual clarity on generative models and applications.
- Correct and responsible use of prompts, tools and datasets.
- Quality of laboratory implementation and mini-project.
- Awareness of ethical, privacy, bias and security issues.

Laboratory Work:

1. Write prompts for summarization, classification, information extraction and content generation; compare output quality.
2. Use a pre-trained transformer model or API for text generation and analyze limitations.
3. Use embeddings and similarity search to retrieve relevant text passages.
4. Build a simple retrieval-augmented question-answering prototype using a small document collection.
5. Demonstrate image generation or image-captioning using an approved tool/model.
6. Create a basic chatbot or AI assistant with input/output guardrails.
7. Prepare a short report on bias, hallucination, privacy or copyright issues in a selected GenAI tool.
8. Complete a mini-project integrating at least one generative model/tool with documentation and presentation.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain foundational concepts, architectures and application areas of generative AI.
CLO2. Analyze neural generative models including attention, transformers, GANs and related approaches.
CLO3. Apply prompt engineering and retrieval-augmented generation techniques for simple tasks.
CLO4. Develop and evaluate a small generative AI application with ethical and responsible-use considerations.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge		CLO1		CLO2		
Procedural Knowledge			CLO3		CLO4	CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	1	0	0	1	0	0	2	0
CLO2	2	2	2	0	0	0	0	0	2	0
CLO3	2	0	0	0	0	0	0	0	2	0
CLO4	2	2	3	0	2	0	0	0	2	2
Average Contribution	**2**	**1**	**1.5**	**0**	**0.5**	**0.25**	**0**	**0**	**2**	**0.5**

(3=High, 2= Moderate, 1= Low, -=No direct correlation)

Text Books:

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
2. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). Dive into deep learning. Cambridge University Press. https://d2l.ai
3. Ganguly, K. (2017). Learning generative adversarial networks: Next-generation deep learning simplified. Packt Publishing.

Reference Books:

1. Jurafsky, D., & Martin, J. H. (n.d.). Speech and language processing (3rd ed. draft). https://web.stanford.edu/~jurafsky/slp3/
2. Chollet, F. (2021). Deep learning with Python (2nd ed.). Manning Publications.
3. Molnar, C. (2019). Interpretable machine learning: A guide for making black box models explainable. Leanpub.

Semester-VI

AEC-1-BCA-601 Soft Skills: Soft Skills

Course Code : AEC-1-BCA-601
Course Title : Soft Skills
Nature of the Course : AEC
Total Credit assigned : 1
Component of the Course : Lecture: 0 Practical: 0 Tutorial: 1

Course Objectives

The main objectives of this course are:

Objective -1. Develop essential communication, presentation and interpersonal skills for academic and professional settings.
Objective -2. Build self-awareness, confidence, time management and goal-setting abilities.
Objective -3. Prepare students for group discussion, interview, workplace communication and professional etiquette.
Objective -4. Encourage ethical behavior, teamwork and lifelong learning attitudes.

Course Content

Unit	Content	L	T	P
I	Self-Management and Personality Development • Meaning and importance of soft skills • Self-awareness, confidence building and emotional intelligence • Goal setting, time management and stress management • Professional attitude, discipline and adaptability • Ethics, responsibility and workplace behavior	0	4	0
II	Communication Skills and Professional Etiquette • Listening, speaking, reading and writing in professional contexts • Verbal, non-verbal and para-linguistic communication • Interpersonal communication and active listening • E-mail etiquette, netiquette, meeting etiquette and telephone/video-call etiquette • Barriers to communication and strategies for improvement	0	4	0
III	Group Interaction, Leadership and Teamwork • Team roles, collaboration and peer learning • Group discussion techniques and evaluation parameters • Negotiation, persuasion and conflict resolution • Leadership styles and decision making in teams • Feedback, empathy and professional relationships	0	4	0
IV	Career Readiness and Presentation Skills • Resume/CV preparation and cover letter writing • Job application and professional profile building • Interview skills, mock interview and HR questions • Oral presentation, body language, voice modulation and slide design • Career planning and lifelong learning	0	3	0

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests / Viva-voce / Presentation, as applicable

Sessional activity:

1. Self-introduction, personal SWOT analysis and goal-setting exercise.
2. Professional e-mail writing and communication etiquette practice.
3. Group discussion and peer feedback session.
4. Mock interview, resume review and short presentation.

Evaluation Criteria:

- Confidence, clarity and appropriateness of communication.
- Ability to participate constructively in group activities.
- Quality of resume, e-mail and presentation materials.
- Professional behavior, punctuality and reflective learning.

Tutorial Work:

1. Prepare a personal SWOT analysis and semester-wise improvement plan.
2. Write a professional e-mail and a cover letter for internship or project application.
3. Participate in one group discussion and submit reflection notes.
4. Prepare and deliver a three-minute individual presentation.
5. Prepare a resume/CV and take part in mock interview practice.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Demonstrate self-management, goal-setting and professional behavior.
CLO2. Communicate effectively through oral, written and digital modes.
CLO3. Participate in teamwork, group discussions and professional interactions.
CLO4. Prepare career documents and perform effectively in presentations and interviews.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge			CLO1, CLO2, CLO3		CLO4	
Metacognitive Knowledge					CLO1	

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	0	0	0	0	0	2	0	2	0	0
CLO2	0	0	0	0	0	2	0	0	0	0
CLO3	0	0	0	0	0	3	2	0	0	0
CLO4	0	0	0	0	0	2	0	0	1	0
Average Contribution	**0**	**0**	**0**	**0**	**0**	**2.25**	**0.5**	**0.5**	**0.25**	**0**

(3=High, 2= Moderate, 1= Low, -=No direct correlation)

Text Books:

1. Kumar, K. B. (2019). Effective communication skills. Khanna Publishing House.
2. Raman, M., & Sharma, S. (2012). Technical communication: Principles and practice (2nd ed.). Oxford University Press.

Reference Books:

1. Guffey, M. E., & Loewy, D. (2018). Essentials of business communication (11th ed.). Cengage Learning.
2. Carnegie, D. (2017). How to win friends and influence people. Simon & Schuster.
3. McCarthy, M., & O'Dell, F. (2017). English vocabulary in use: Upper-intermediate (4th ed.). Cambridge University Press.

Semester-VI

SEC-4-BCA-601: Major Project - Phase II: Implementation and Evaluation

Course Code	**: SEC-4-BCA-601**
Course Title	**: Major Project - Phase II: Implementation and Evaluation**
Nature of the Course	**: SEC**
Total Credit assigned	**: 4**
Component of the Course	**: Lecture: 0 Practical: 8 Tutorial: 0**

Course Objectives

The main objectives of this course are:

Objective -1. Enable students to implement the major project planned in the fifth semester.
Objective -2. Develop practical ability in software development, data analysis, testing, documentation and deployment.
Objective -3. Encourage professional project management, teamwork, version control and ethical handling of data/software.
Objective -4. Evaluate students through project report, demonstration, presentation and viva-voce.

Course Content

Project Work - Phase II:

- The sixth-semester major project component shall be treated as the implementation, testing, documentation, presentation and final evaluation phase of the project initiated in the fifth semester.
- Students shall implement the approved project using suitable programming tools, databases, frameworks, datasets, algorithms, platforms or research methods, as applicable.
- The work shall include module-wise implementation, testing/validation, result analysis, improvement based on guide feedback, and preparation of deployable artifacts or demonstrable outputs.
- Students shall prepare the final project report including problem statement, objectives, methodology, design, implementation, testing, results, limitations, future scope and references.
- Evaluation shall be based on project report, demonstration, presentation, viva-voce, originality, technical quality, documentation, teamwork/professional conduct and compliance with departmental guidelines.

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests / Viva-voce / Presentation, as applicable

Sessional activity:

1. Weekly progress demonstration before guide.
2. Mid-term review of implementation and testing status.
3. Submission of draft project report and source-code/documentation artifacts.
4. Final project demonstration and viva-voce.

Evaluation Criteria:

- Problem relevance, originality and implementation quality.
- Functionality, testing, usability, security and documentation quality.
- Consistency of progress, team contribution and professional conduct.
- Quality of final report, presentation, demonstration and viva-voce.

Project Work:

1. Develop the approved major project using appropriate programming language, tools, dataset or platform.
2. Maintain version-controlled source code and project diary.
3. Test individual modules and integrated system with suitable test cases.
4. Prepare screenshots, diagrams, database schema, API documentation or experimental outputs.
5. Prepare final project report in prescribed format.
6. Present and demonstrate the project before the evaluation committee.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Implement a computing project according to approved requirements and design.
CLO2. Apply appropriate tools, algorithms, databases, testing methods and documentation practices.
CLO3. Evaluate project functionality, usability, performance, limitations and future scope.
CLO4. Present and defend the completed project through report, demonstration and viva-voce.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge				CLO3	CLO4	CLO1,CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	0	3	3	0	0	0	2	3	0	0
CLO2	3	2	3	3	0	2	2	3	1	0
CLO3	0	2	3	0	0	0	2	3	0	0
CLO4	0	0	3	0	0	2	2	3	0	0
Average Contribution	**0.75**	**1.75**	**3**	**0.75**	**0**	**1**	**2**	**3**	**0.25**	**0**

(3=High, 2= Moderate, 1= Low, -=No direct correlation)

Text Books:

1. Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
2. Sommerville, I. (2016). Software engineering (10th ed.). Pearson.

Reference Books:

1. Schwalbe, K. (2019). Information technology project management (9th ed.). Cengage Learning.
2. Dennis, A., Wixom, B. H., & Roth, R. M. (2018). Systems analysis and design (7th ed.). Wiley.
3. IEEE Computer Society. (2014). Guide to the software engineering body of knowledge (SWEBOK guide) (Version 3.0). IEEE.

SEMESTER –III

Course Code :**DSE-3-BCA-301**
Course Title :**Feature Engineering**
Nature of the Course :**DSE**
Total Credit assigned :**3**
Component of the Course :**Lecture: 2 Practical: 1 Tutorial: 0**
Distribution of Marks :**End Semester: 28 Practical:20 In Semester: 12**

Course Objectives:

CO1. To understand the importance of features in machine learning and learn basic preprocessing techniques such as handling missing data, scaling, and normalization.
CO2. To explore advanced feature engineering techniques, including feature extraction, selection, and reduction, and apply methods like PCA and encoding for different types of data.

Prerequisite: Basic knowledge of data analytics/machine learning and familiarity with any programming language

Course content:

Unit	Content	Marks	L	P	T
I	Introduction to Feature Engineering Introduction to Data and Features: Importance of Features in Machine Learning. Data types and features: Numerical, Categorical, Ordinal, Discrete, Continuous, Interval and Ratio.Basic Feature Preprocessing: Handling Missing Data, Data Cleaning, Feature Scaling, Normalization, and Transformation.	13	15	14	0
II	Feature Engineering Techniques Techniques for Numerical Data: Binning and Discretization, Polynomial and Interaction Features.Categorical Data Techniques: One Hot Encoding, Label Encoding. Feature extraction vs. feature selection, Steps in feature selection. Feature Selection Methods: Filter, Wrapper, and Hybrid. Feature Reduction: Introduction and application of Principal Components Analysis.	15	15	16	0
Total		28	30	30	0

Mode of In-Semester assessment (12 Marks):
A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

1. **Data Preprocessing Exercise:** Students will work with a dataset, handling missing data, scaling features, and performing normalization/standardization.
2. **Feature Encoding Task:** Students will practice encoding categorical variables using One-Hot and Label Encoding methods.
3. **Feature Engineering Workshop:** Students will create polynomial features and interaction terms, then evaluate their effect on model performance.

4. **Principal Component Analysis (PCA) Exercise:** Students will reduce the dimensionality of a dataset using PCA and compare the results before and after reduction.
5. **Feature Selection Methods:** Students will implement and compare filter, wrapper, and hybrid feature selection methods on a given dataset.

Suggested list of LAB Assignments (not limited to):

Prerequisite: Knowledge of Python Programming language

LAB Experiments

The lab experiments can be implemented in Python using relevant libraries such as numpy, pandas, sklearn, nltk, matplotlib, and seaborn. Kaggle datasets, public repositories (e.g., UCI, Machine Learning etc.), or generated datasets can be used for conducting the experiments. Experiments may be conducted on numerical, image, or time-series datasets.

1. Handle missing values in column(s) of a dataset. For example, fill missing values with the mean/median/mode of the columns such as ‘Age’, ‘Height’, ‘Weight’, ‘Grade’ for a dataset.
2. Clean a dataset by identifying and removing invalid data entries. For example, a dataset having columns 'Name', ‘Gender’ and 'Age' where 'Name' contains 'invalid data'.
3. Scale numerical features using Min-Max normalization for a dataset with columns like ‘Height', ‘Weight’.
4. Perform exploratory data analysis and visualize data distributions using histograms and boxplots.
5. Compute and visualize the correlation matrix of a dataset with 2 or more columns.
6. Bin numerical data into discrete intervals for a dataset with a column containing numerical values.
7. Create polynomial and interaction features from numerical data in a dataset with two columns.
8. Apply logarithmic transformation to skewed numerical features in a dataset with column 'Distance'.
9. Perform one-hot encoding on categorical features in a dataset with column 'Category' containing categorical values. The distinct values in the Category feature are [Good, Better, Best] and Gender [Male, Female].
10. Preprocess text data (tokenization) for a dataset with a column 'Text'.
11. Preprocess text data (stemming) for a dataset with a column 'Text'.
12. Preprocess text data (lemmatization) for a dataset with a column 'Text'.
13. Convert text data into a Bag-of-Words representation for a dataset with a column 'Text'.
14. Apply TF-IDF transformation to text data for a column 'Text'.
15. Perform image augmentation (resizing, normalization, rotation, translation) for a set of images.
16. Perform image augmentation resizing for a set of images.
17. Perform image augmentation normalization for a set of images.
18. Perform image augmentation rotation for a set of images.
19. Perform image augmentation translation for a set of images.
20. Decompose a time series into trend, seasonal, and residual components for a dataset with a column 'TimeSeries'.
21. Perform Principal Component Analysis (PCA) on a dataset and visualize the first two principal components.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

1. **CLO1:** Preprocess and clean data, handling missing values, and scaling features for machine learning applications.
2. **CLO2:** Apply various feature engineering techniques, including encoding categorical data, creating interaction features, and performing feature selection and reduction (e.g., PCA).

Cognitive Map of Course Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge	CO_1	CO_2	CO_1, CO_2			
Conceptual Knowledge	CO_1	CO_2				
Procedural Knowledge			CO_2			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	3	2	2	1	1	0	0	0	0
CLO2	3	3	3	2	2	1	0	0	0	0
Average Contribution	3.0	3.0	2.5	2.0	1.5	1.0	0.0	0.0	0.0	0.0

(3=High, 2= Moderate 1= Low)

Text Books

1. Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.".
2. Kalita, J. K., Bhattacharyya, D. K., & Roy, S. (2023). *Fundamentals of data science: theory and practice*. Elsevier.

Reference Books:

1. Duda, R. O., & Hart, P. E. (2006). *Pattern classification*. John Wiley & Sons.
2. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. ISBN 978-0123814791. *Chapter*, *10*, 443-460.
3. Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.
4. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking*. " O'Reilly Media, Inc.".
5. Galli, S. (2022). *Python feature engineering cookbook: over 70 recipes for creating, engineering, and transforming features to build machine learning models*. Packt Publishing Ltd.
6. Nielsen, A. (2019). *Practical time series analysis: Prediction with statistics and machine learning*. O'Reilly Media..
7. Francois, C. (2018). Deep learning with Python..

SEMESTER–IV

DSE-3-BCA-401: Introduction to Machine Learning

Course Code : DSE-3-BCA-401
Course Title : Introduction to Machine Learning
Nature of the Course : DSE
Total Credit assigned : 3
Component of the Course : Lecture:1 Practical: 2 Tutorial: 0
Distribution of Marks : End Semester: 28 Practical: 40 In Semester: 12

Course Objectives:

CO1. To introduce the fundamental concepts, types, and workflow of machine learning, along with model evaluation techniques.
CO2. To provide practical understanding of supervised and unsupervised learning algorithms, including regression, classification, clustering, and their real-world applications.

Course Content:

Unit	Content	Marks	L	P	T
I	**Introduction to Machine Learning** *Introduction:* Definition, History and Application of Machine Learning, *Types of Machine Learning*: Supervised, Unsupervised, Semi-Supervised, and Reinforcement Learning. Labelled and Unlabelled Dataset. *Supervised Learning Tasks*: Regression vs. Classification, *Learning Framework*: Training, Validation and Testing of ML models. *Performance Evaluation Parameters*: Confusion matrix, Accuracy, Precision, Recall, F1Score, and AUC.	13	7	18	0
II	**Supervised Learning and Unsupervised Learning** *Regression*: Linear and non-linear Regression, Logistic Regression. *Classification*: Naïve Bayes, K-Nearest Neighbors, Decision Trees. *Linearmodel*: Introduction to Artificial Neural Networks, Perceptron Learning Algorithm, Single Layer Perceptron, Introduction to Support Vector Machine for linearly separable data. *Clustering*: K-Means, Hierarchical Clustering, DBSCAN, Clustering Validation Measures. *ML Applications:* Ethical Considerations in Machine Learning, Case study and Real-world Applications.	15	8	22	0
		28	15	40	

Mode of In-Semester assessment (12 Marks):
A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional Activities:

- **ML in Real Life**– Group discussion on ML applications (e.g., Netflix, fraud detection).
- **Dataset Activity**– Identify labelled vs unlabelled datasets and match them with ML types.
- **Mini Project: Regression & Classification**– Apply linear/logistic regression and one classifier (e.g., KNN) on real data using Python/Colab.
- **Clustering Demo**– Perform K-Means clustering on a small dataset and interpret results.
- **Model Evaluation Task**– Given confusion matrix data, compute accuracy, precision, recall, F1-score.

Suggested list of LAB Assignments (not limited to):

1. Implement linear regression on a dataset and visualize the regression line.
2. Implement logistic regression on a binary classification dataset and plot the decision boundary.
3. Implement and evaluate the performance of Decision tree ID3/Cart classifier for any given dataset.
4. Implement and evaluate the performance of the Naive Bayes Classifier on a given dataset.
5. Build and evaluate a random forest classifier using a numerical dataset.
6. Implement a support vector machine for linearly separable classes and visualize the margins and decision boundary.
7. Implement K-Means clustering on a point dataset and visualize and evaluate the clusters.
8. Implement hierarchical clustering on a dataset and plot the dendrogram.
9. Implement DBSCAN clustering on a dataset and visualize and evaluate the clusters.
10. Perform Principal Components Analysis (PCA) and apply any one or more classifiers to show the performance variation with or without feature reduction.
11. Build a single layer perceptron model to classify AND, OR, and XOR problems (may use TensorFlow/Keras) and visualize their decision boundaries. Also evaluate its performance.
12. Demonstrate the concept of boosting using the AdaBoost algorithm.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO-1. Explain types of machine learning, differentiate between regression and classification, and evaluate models using standard performance metrics.
CLO-2. Apply ML algorithms like regression, classification, and clustering, and analyze case studies while understanding ethical implications.

Cognitive Map of Course Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge	CO_1					
Conceptual Knowledge	CO_1	CO_2	CO_2	CO_2		
Procedural Knowledge		CO_1	CO_2	CO_1		
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	0	0	0	0	0	0	2	0
CLO2	2	3	3	0	0	0	0	0	1	0
Average Contribution	2	2.5	1.5	0	0	0	0	0	1.5	0

(3=High, 2= Moderate 1= Low)

Text Books:

1. Vijayalakshmi, M., & Sridhar, S. (2021). Machine Learning.
2. Kalita, J. K., Bhattacharyya, D. K., & Roy, S. (2023). *Fundamentals of data science: theory and practice*. Elsevier.

Reference Books:

1. Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data*. Cambridge university press.
2. Duda, R. O., & Hart, P. E. (2006). *Pattern classification*. John Wiley & Sons..
3. Haykin, S. (2009). *Neural networks and learning machines, 3/E*. Pearson Education India..
4. Chollet, F., & Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster..
5. Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.
6. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning, 1 MIT press. *deeplearningbook. org/contents/numerical. htm (02.07. 2023)*..
7. Géron, A. (2017). Hands-On machine learning with Scikit-Learn and TensorFlow: concepts, tools. *and Techniques to Build Intelligent Systems*..

Semester-V

DSE-5-BCA-501: Neural Network

Course Code : DSE-5-BCA-501
Course Title : Neural Network
Nature of the Course : DSE / Professional Elective
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives:

The main objectives of this course are:

Objective -1. Introduce the principles of artificial neural networks, perceptron models, activation functions, and training methods.
Objective -2. Familiarize students with dense, deep, convolutional, recurrent, and graph neural-network architectures.
Objective -3. Develop the ability to select suitable neural-network architectures for different data types and tasks.
Objective -4. Enable students to analyze training issues, compare model performance, and evaluate alternative network designs.

Course Content

Unit	Content	L	P	T
I	**Fundamentals of Neural Networks** • Artificial neurons and layers; biological and artificial neural networks • Perceptron/MLP architecture; activation functions; loss functions • Forward propagation, backward propagation, automatic differentiation, regularization and training issues	9	12	0
II	**Dense and Deep Neural Networks** • Dense/fully connected networks and deep neural networks • Gradient-based learning; model capacity; overfitting and underfitting • Generalization and practical training tricks for effective design	9	12	0
III	**Convolutional Neural Networks** • CNN concepts for image-like data • Convolution, feature maps and pooling • Architectures including LeNet-5, AlexNet, Inception and ResNet; recognition and classification applications	9	12	0
IV	**Sequence Neural Networks** • Recurrent neural networks for ordered data • LSTM and GRU models; memory in sequential learning • Common sequence-modelling applications	9	12	0
V	**Graph Neural Networks** • Graph-structured data and graph neural network concepts • Matching network architecture to data type and task	9	12	0
	Total	**45**	**60**	**0**

Mode of In-Semester Assessment

A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Implement and train an MLP for classification or regression.
- Train a small CNN for image classification.
- Experiment with RNN/LSTM/GRU for sequence data.
- Compare neural-network architectures on a selected dataset.

Evaluation Criteria:

- Conceptual understanding and correctness of solution.
- Appropriate use of tools, models, algorithms and datasets.
- Quality of implementation, analysis and interpretation.
- Clarity of documentation, presentation and viva voce.

Laboratory Work

1. Implement and train a multilayer perceptron for a classification problem.
2. Develop a CNN model for basic image classification.
3. Implement an RNN/LSTM/GRU model for simple sequence prediction.
4. Compare training accuracy, validation accuracy, loss and computational cost of different architectures.

Expected Course Learning Outcomes

By the end of the course, students will be able to:

CLO1. Explain major neural-network architectures and their appropriate use cases.
CLO2. Design suitable neural-network models for classification, regression, sequence or image-based tasks.
CLO3. Analyze computational and training challenges such as overfitting, underfitting and generalization.
CLO4. Compare the performance of alternative neural-network designs on selected datasets.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO1				
Conceptual Knowledge					CLO3	
Procedural Knowledge				CLO4		CLO2
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	0	0	0	1	0	0	2	0
CLO2	2	3	3	0	0	0	0	0	0	0
CLO3	2	2	0	0	0	0	0	0	2	0
CLO4	2	3	3	0	0	0	0	0	1	0
Average Contribution	2	2	1.5	0	0	0.25	0	0	1.25	0

(3=High, 2= Moderate, 1= Low, -=No Direct Correlation)

Suggested Readings

1. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). Dive into deep learning. Cambridge University Press. https://d2l.ai
2. Geron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
4. Haykin, S. (2009). Neural networks and learning machines (3rd ed.). Pearson.

Semester-V

DSE-5-BCA-502: Digital Image Processing

Course Code : DSE-5-BCA-502
Course Title : Digital Image Processing
Nature of the Course : DSE / Professional Elective
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives:

The main objectives of this course are:

Objective -1. Introduce digital image representation, sampling, quantization, spatial resolution and intensity-level concepts.
Objective -2. Develop understanding of image enhancement, restoration, filtering, morphological processing and compression techniques.
Objective -3. Enable students to apply segmentation and basic computer-vision methods to image data.
Objective -4. Train students to use image-processing tools for practical image analysis and interpretation.

Course Content

Unit	Content	L	P	T
I	**Image Fundamentals** • Visual perception and digital image representation • Image sampling and quantization • Pixels, spatial resolution and intensity levels	9	12	0
II	**Image Enhancement and Restoration** • Intensity transformations and nonlinear enhancement methods • Histogram equalization and spatial filtering • Wiener filtering and restoration concepts	9	12	0
III	**Colour and Morphological Processing** • Colour spaces and colour image processing • Erosion, dilation, opening and closing • Hit-or-miss transform, thinning and shape decomposition	9	12	0
IV	**Segmentation and Compression** • Edge detection, thresholding and region-based segmentation • Fundamentals of image compression • Lossless coding, predictive coding and transform coding • Image formats such as JPEG, PNG encoding	9	12	0
V	**Computer Vision Extensions** • Cameras and projection models; clustering • Stereo-based shape reconstruction; object and scene recognition • Face detection and human motion categorization	9	12	0
	Total	**45**	**60**	**0**

Mode of In-Semester Assessment

A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Perform histogram equalization and spatial filtering.
- Apply morphological operations on binary and grayscale images.
- Implement edge detection and thresholding-based segmentation.
- Use OpenCV for basic image-processing and vision tasks.

Evaluation Criteria:

- Conceptual understanding and correctness of solution.
- Appropriate use of tools, models, algorithms and datasets.
- Quality of implementation, analysis and interpretation.
- Clarity of documentation, presentation and viva voce.

Laboratory Work

1. Perform image sampling, quantization and intensity transformation experiments.
2. Apply histogram equalization, smoothing and sharpening filters to sample images.
3. Implement morphological operations and segmentation on binary/grayscale images.
4. Use OpenCV/Python to perform edge detection, object detection and simple vision tasks.

Expected Course Learning Outcomes

By the end of the course, students will be able to:

CLO1. Explain image sampling, quantization and digital image representation.
CLO2. Apply enhancement, restoration, segmentation and compression techniques to images.
CLO3. Use image-processing tools such as OpenCV for practical image analysis.
CLO4. Relate digital image-processing techniques to basic computer-vision applications.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge		CLO1				
Conceptual Knowledge				CLO4		
Procedural Knowledge			CLO2,CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	0	0	0	2	0	0	2	0
CLO2	2	0	0	0	0	0	0	0	0	0
CLO3	2	2	0	3	0	0	0	0	1	0
CLO4	2	0	2	0	0	0	0	0	0	0

Average Contribution	2	0.5	0.5	0.75	0	0.5	0	0	0.75	0

(3=High, 2= Moderate, 1= Low, -=No Direct Correlation)

Suggested Readings

1. Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing (4th ed.). Pearson.
2. Szeliski, R. (2010). Computer vision: Algorithms and applications. Springer.
3. Bradski, G., & Kaehler, A. (2008). Learning OpenCV. O'Reilly Media.
4. Pratt, W. K. (2007). Digital image processing (4th ed.). Wiley.

Semester-V

DSE-5-BCA-503: Natural Language Processing

Course Code : DSE-5-BCA-503
Course Title : Natural Language Processing
Nature of the Course : DSE / Professional Elective
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives

The main objectives of this course are:

Objective -1. Introduce text preprocessing, representation, tokenization, n-grams, TF-IDF and word-vector techniques.
Objective -2. Develop skill in collecting, cleaning and preparing text data using NLP tools and corpora.
Objective -3. Enable students to perform core NLP tasks such as POS tagging, NER, sentiment analysis and topic modelling.
Objective -4. Familiarize students with advanced NLP applications including machine translation, question answering and multilingual retrieval.

Course Content

Unit	Content	L	P	T
I	**Text Pre-processing and Representation** • Tokenization, n-grams and TF-IDF • Feature extraction, word vectors and document models • Word-level and document-level representations	9	12	0
II	**Text Data Collection and Tools** • Data collection using APIs, social media and web scraping • NLP software/toolkits such as NLTK • Preparation of text corpora	9	12	0
III	**Core NLP Tasks** • Part-of-speech tagging and named entity recognition • Sentiment analysis and topic modelling • Stylometry and text classification concepts	9	12	0
IV	**Text Visualization and Clustering** • Text visualization using dendrograms, PCA and text plots • Document clustering and topic discovery • Analysis of clustered text documents	9	12	0
V	**Advanced NLP Applications** • Machine translation and question answering • Multilingual settings and cross-lingual information retrieval • Practical NLP system/project development	9	12	0
	Total	**45**	**60**	**0**

Mode of In-Semester Assessment

A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Build a text preprocessing pipeline.
- Perform POS tagging, NER and sentiment analysis.
- Visualize and cluster text documents.
- Create a small NLP project using an available toolkit.

Evaluation Criteria:

- Conceptual understanding and correctness of solution.
- Appropriate use of tools, models, algorithms and datasets.
- Quality of implementation, analysis and interpretation.
- Clarity of documentation, presentation and viva voce.

Laboratory Work

1. Create a text-cleaning pipeline including tokenization, stop-word removal and vectorization.
2. Perform POS tagging and named entity recognition using an NLP toolkit.
3. Develop a sentiment-analysis or topic-modelling application.
4. Visualize text data and cluster documents using suitable features.

Expected Course Learning Outcomes

By the end of the course, students will be able to:

CLO1. Explain core NLP representations such as n-grams, TF-IDF and word vectors.
CLO2. Collect, clean and prepare text data from digital sources.
CLO3. Perform POS tagging, NER, sentiment analysis, topic modelling and text visualization.
CLO4. Apply ML/NLP tools to build a working NLP application.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge		CLO1,				
Procedural Knowledge			CLO2,CLO3			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	0	0	0	2	0	0	2	0
CLO2	0	2	0	0	0	0	0	0	1	0
CLO3	2	2	2	2	0	0	0	0	0	0
CLO4	2	0	3	3	0	0	0	0	1	0
Average Contribution	1.5	1	1.25	1.25	0	0.5	0	0	1	0

(3=High, 2= Moderate, 1= Low, -=No Direct Correlation)

Suggested Readings

1. Jurafsky, D., & Martin, J. H. (n.d.). Speech and language processing (3rd ed. draft).
2. Eisenstein, J. (2019). Introduction to natural language processing. MIT Press.
3. Rao, D., & McMahan, B. (2019). Natural language processing with PyTorch. O'Reilly Media.
4. Manning, C. D., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT Press.

Semester-VI

Course Code : DSE-5-BCA-601
Course Title : Deep Learning for Computer Vision
Nature of the Course : DSE / Professional Elective
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives

The main objectives of this course are:

Objective -1. Introduce CNN-based deep-learning approaches for computer-vision tasks.
Objective -2. Familiarize students with advanced CNN architectures, transfer learning and efficient vision models.
Objective -3. Develop understanding of sequence models, attention and image-captioning methods.
Objective -4. Enable students to apply generative adversarial networks for synthetic-data generation and vision applications.

Course Content

Unit	Content	L	P	T
I	**Convolutional Neural Networks** • Computer vision and CNN architecture • Convolution, pooling and fully connected layers • Activation functions, batch normalization, parameters and hyperparameters	10	12	0
II	**Recurrent Neural Networks** • RNN architecture and ordered sequences • Sequential text generation and memory in deep-learning models • Applications of RNNs	10	12	0
III	**Attention and Image Captioning** • Attention in NLP and computer vision • Combining CNN and RNN for image-captioning models • LSTM for caption generation	12	16	0
IV	**Generative Adversarial Networks** • GAN concepts, discriminator and generator • Activation functions and binary cross-entropy loss • Conditional GANs, controllable generation and real-life GAN applications	13	16	0
	Total	**45**	**60**	**0**

Mode of In-Semester Assessment

A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Apply a basic CNN for image classification.
- Use transfer learning with an advanced CNN.
- Apply recurrent neural networks to ordered data.
- Use LSTM for image captioning and GANs for synthetic-data generation.

Evaluation Criteria:

- Conceptual understanding and correctness of solution.
- Appropriate use of tools, models, algorithms and datasets.
- Quality of implementation, analysis and interpretation.
- Clarity of documentation, presentation and viva voce.

Laboratory Work

1. Train a basic CNN for image classification.
2. Apply transfer learning using a pretrained CNN architecture.
3. Implement a simple RNN/LSTM-based sequence model.
4. Build a mini image-captioning or GAN-based demonstration.

Expected Course Learning Outcomes

By the end of the course, students will be able to:

CLO1. Explain deep-learning models used for computer-vision tasks.
CLO2. Apply CNN and transfer-learning methods to image data.
CLO3. Use attention and LSTM concepts for image-captioning or sequence-based vision tasks.
CLO4. Generate or analyze synthetic image data using GAN concepts.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge		CLO1,				
Procedural Knowledge			CLO2, CLO3			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	1	0	0	1	0	0	2	0
CLO2	2	2	0	0	0	0	0	0	1	0
CLO3	2	0	0	0	0	0	0	0	1	0
CLO4	2	2	0	0	0	0	0	0	1	0
Average Contribution	2	1.5	0.25	0	0	0.25	0	0	1.25	0

(3=High, 2= Moderate, 1= Low, -=No Direct Correlation)

Suggested Readings

1. Solem, J. E. (2012). Programming computer vision with Python. O'Reilly Media.
2. Planche, B., & Andres, E. (2019). Hands-on computer vision with TensorFlow 2. Packt Publishing.
3. Ganguly, K. (2017). Learning generative adversarial networks. Packt Publishing.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Semester-VI

Course Code : DSE-5-BCA-602
Course Title : Predictive Analysis
Nature of the Course : DSE / Professional Elective
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Course Objectives

The main objectives of this course are:

Objective -1. Introduce data products, predictive modelling strategy and predictive-analysis workflows.
Objective -2. Develop skills in reading, cleaning and processing structured data using Python.
Objective -3. Familiarize students with Python libraries for visualization, data collection and matrix processing.
Objective -4. Enable students to train, validate and evaluate predictive models using diagnostics and performance metrics.

Course Content

Unit	Content	L	P	T
I	**Data Products and Strategy** • Meaning of data product and examples in enterprise settings • Developing a data-product strategy for predictive tasks • Predictive-analysis workflow and problem framing	9	12	0
II	**Reading and Processing Data in Python** • Reading CSV and JSON files • Filtering, cleaning and processing structured data • Processing text, strings, times and dates	9	12	0
III	**Python Libraries and Toolkits** • Matrix processing and NumPy • Data visualization using Matplotlib • Data collection using urllib and BeautifulSoup	9	12	0
IV	**Gradient Descent and Classification** • Classification in Python • Training and testing concepts • Gradient descent in Python and TensorFlow	9	12	0
V	**Diagnostics and Model Evaluation** • Regression diagnostics and meaningful predictive modelling • Overfitting, underfitting, validation and regularization • Classification diagnostics: accuracy, error, precision, recall and ranking of classifiers	9	12	0
	Total	**45**	**60**	**0**

Mode of In-Semester Assessment

A. Sessional Activity
B. Sessional Tests: (two Sessional Tests)

Sessional activity:

- Process data for dates, times and strings.
- Use Python libraries for visualization.
- Develop a basic predictive or TensorFlow model.
- Perform regression and classification diagnostics.

Evaluation Criteria:

- Conceptual understanding and correctness of solution.
- Appropriate use of tools, models, algorithms and datasets.
- Quality of implementation, analysis and interpretation.
- Clarity of documentation, presentation and viva voce.

Laboratory Work

1. Read, clean and preprocess CSV/JSON datasets using Python.
2. Create exploratory visualizations using Matplotlib or similar tools.
3. Train and test regression/classification models.
4. Evaluate models using accuracy, precision, recall and validation diagnostics.

Expected Course Learning Outcomes

By the end of the course, students will be able to:

CLO1. Prepare structured data for predictive modelling.
CLO2. Use Python tools and libraries for predictive analysis.
CLO3. Train and evaluate predictive models using suitable datasets.
CLO4. Diagnose model fit and interpret predictive-modelling evaluation metrics.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	Remember	Understand	Apply	Analyse	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge					CLO4	
Procedural Knowledge			CLO1, CLO2, CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	2	0	0	0	0	0	1	0
CLO2	2	2	0	3	0	0	0	0	1	0
CLO3	2	2	2	0	0	0	0	0	2	0
CLO4	2	2	2	0	0	0	0	0	0	0
Average Contribution	2	2	1.5	0.75	0	0	0	0	1	0

(3=High, 2= Moderate, 1= Low, -=No Direct Correlation)

Suggested Readings

1. Abbott, D. (2014). Applied predictive analytics. Wiley.
2. Provost, F., & Fawcett, T. (2013). Data science for business. O'Reilly Media.
3. Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer.
4. James, G., Witten, D., Hastie, T., &Tibshirani, R. (2021). An introduction to statistical learning (2nd ed.). Springer.

Semester-III

Course Code : DSE-3-BCA-301-D
Course Title : Basics of Data Analytics using Spreadsheet
Nature of the Course : DSE
Stream : Data Science
Total Credit assigned : 3
Component of the Course : Lecture: 1 Practical: 2 Tutorial: 0

Prerequisite : Basic arithmetic, percentages, averages, basic algebra, and familiarity with computer file handling.

Course Objectives

The main objectives of this course are:

Objective -1. Understand basic concepts, types, and applications of data analytics.
Objective -2. Use spreadsheet software for data entry, cleaning, transformation, statistical analysis, and decision support.
Objective -3. Develop simple analytical models and communicate insights through charts, PivotTables, and dashboards.

Course Content

Unit	Content	L	P	T
I	Introduction to Data AnalyticsData, information, and knowledge; structured, semi-structured, and unstructured data; meaning and importance of data analytics; descriptive, diagnostic, predictive, and prescriptive analytics; applications in education, finance, marketing, operations, public administration, and social sectors.	4	15	0
II	Data Collection, Cleaning, Ethics, and Case StudiesData sources and formats; manual and automated collection; CSV, text, web, and database data; data cleaning, transformation, handling missing values and outliers; ethical use of data; case studies from finance, marketing, operations, and education.	4	15	0
III	Spreadsheet-Based AnalysisWorkbook, worksheet, cells, ranges, relative/absolute/mixed references; basic functions such as SUM, AVERAGE, MIN, MAX, ROUND; descriptive statistics; frequency distributions; histograms; Data Analysis ToolPak; logical functions; lookup and reference functions such as VLOOKUP, HLOOKUP, INDEX, MATCH, SUMIFS, COUNTIFS, and AVERAGEIFS.	4	15	0
IV	Visualization and Dashboarding in SpreadsheetChart types and chart selection; bar, line, pie, scatter, combo, and dual-axis charts; PivotTables and PivotCharts; slicers and timelines; dashboard components; conditional formatting; presenting spreadsheet-based analytical reports.	3	15	0
	Total	**15**	**60**	**0**

Mode of In-Semester assessment:

A. Sessional Activity
B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Create and format workbooks, worksheets, tables, and ranges.
2. Import CSV/text/web data and clean duplicate, missing, and inconsistent entries.
3. Compute descriptive statistics and create histograms.
4. Use IF, AND, OR, IFERROR, lookup, text, and aggregation functions.
5. Prepare PivotTables, PivotCharts, slicers, timelines, and an interactive dashboard.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain different types of data and analytics used in decision making.
CLO2. Clean and prepare data using spreadsheet tools.
CLO3. Apply spreadsheet functions for descriptive and decision-oriented analysis.
CLO4. Create charts, PivotTables, and dashboards to communicate insights.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge		CLO1				
Conceptual Knowledge						
Procedural Knowledge			CLO2,CLO3			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	0	2	0	1	0	0	2	0
CLO2	2	2	0	2	0	0	0	0	1	0
CLO3	2	2	0	3	0	0	0	0	1	0
CLO4	2	2	3	2	0	1	0	0	1	0
Average Contribution	**2**	**2**	**0.75**	**2.25**	**0**	**0.5**	**0**	**0**	**1.25**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Jose, J. (2024). Beginner's guide for data analysis using R programming. Khanna Publishing House.
2. Jain, V. K. (2024). Data analytics. Khanna Book Publishing Company.
3. Nelson, S. L., & Nelson, E. C. (2016). Excel data analysis for dummies (3rd ed.). Wiley.
4. Middleton, M. R. (2004). Data analysis using Microsoft Excel (3rd ed.). Thomson/Brooks/Cole.

Reference Books

1. Alexander, M., Kusleika, R., & Walkenbach, J. (2018). Excel 2019 Bible. Wiley.
2. Ragsdale, C. T. (2015). Spreadsheet modeling and decision analysis: A practical introduction to business analytics. Cengage Learning.
3. Trivedi, M. C. (2025). Data science and data analytics using Python. Khanna Publishing House.

Semester-IV

DSE-3-BCA-401-D: Data Visualization

Course Code :DSE-3-BCA-401-D
Course Title :Data Visualization
Nature of the Course :DSE
Stream :Data Science
Total Credit assigned :3
Component of the Course :Lecture: 1 Practical: 2 Tutorial: 0

Prerequisite : Basic knowledge of tables, data types, data analysis concepts, and computer operations.

Course Objectives

The main objectives of this course are:

Objective -1. Understand fundamentals and importance of data visualization.
Objective -2. Select suitable visual encodings for numerical, categorical, temporal, and geographical data.
Objective -3. Apply storytelling, dashboard design, and Power BI-based report creation.

Course Content

Unit	Content	L	P	T
I	Introduction to Data Visualization:Definition and importance of visualization; visualization in decision making; data types: numerical, categorical, temporal, geographical; visualization process: collection, exploration, analysis, visualization, interpretation; challenges and limitations.	4	15	0
II	Visualization Tools and Data Storytelling:Overview and comparison of Excel, Tableau, Power BI, and Python visualization libraries; tool selection and use cases; narrative, context, audience, dashboard layout, interactivity, and visual storytelling principles.	4	15	0
III	Designing Effective Visualizations: Principles of good visualization design; graphical perception; chart selection; use of color, scale, shape, labels, legends, and annotations; avoiding misleading visualizations; importance of data modeling in visualization.	4	15	0
IV	Power BI Reports and Dashboards: Data import and transformation in Power BI; data model relationships; measures; bar, line, pie, matrix, map, and KPI visuals; slicers and filters; report publishing; sharing dashboards; refresh schedules and permissions.	3	15	0
	Total	**15**	**60**	**0**

Mode of In-Semester assessment:
A. Sessional Activity

B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Connect Power BI to Excel/CSV datasets and transform data.
2. Build basic charts and maps with slicers and filters.
3. Design a multi-page report using a meaningful layout and theme.
4. Create a dashboard with KPIs and drill-down features.
5. Publish a report to Power BI service and configure sharing/refresh.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Choose appropriate visualizations for different data and analytical purposes.
CLO2. Build effective and ethical dashboards using visual design principles.
CLO3. Use Power BI for interactive visualization and report sharing.
CLO4. Interpret and communicate visual insights clearly for decision making.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge				CLO1,CLO4	CLO2	
Procedural Knowledge			CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	0	2	0	0	0	0	1	0
CLO2	2	3	3	3	2	0	0	0	1	0
CLO3	2	2	0	3	0	2	0	0	1	0
CLO4	2	2	0	2	0	1	0	0	1	0
Average Contribution	**2**	**2.25**	**0.75**	**2.5**	**0.5**	**0.75**	**0**	**0**	**1**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Knaflic, C. N. (2015). Storytelling with data: A data visualization guide for business professionals. Wiley.

2. Tufte, E. R. (2001). The visual display of quantitative information (2nd ed.). Graphics Press.

Reference Books

1. Healy, K. (2019). Data visualization: A practical introduction. Princeton University Press.

2. Ferrari, A., & Russo, M. (2017). Analyzing data with Power BI and Power Pivot for Excel. Microsoft Press.

3. Knight, D., Knight, B., Pearson, M., &Quintana, M. (2018). Microsoft Power BI complete reference. Packt Publishing.

4. Trivedi, M. C. (2025). Data science and data analytics using Python. Khanna Publishing House.

Semester-V

DSE-5-BCA-501-D: Introduction to Data Science

Course Code : DSE-5-BCA-501-D
Course Title : Introduction to Data Science
Nature of the Course : DSE
Stream : Data Science
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Prerequisite : Python programming, basic statistics, and basic database concepts.

Course Objectives

The main objectives of this course are:

Objective -1. Introduce the data science lifecycle, tools, and application areas.
Objective 2. Develop ability to collect, manage, clean, explore, and visualize data.
Objective -3. Use Python-based libraries for basic data analysis and prediction.
Objective -4. Discuss ethics, privacy, security, and responsible data use.

Course Content

Unit	Content	L	P	T
I	Data Science Overview and Tools: Meaning and scope of data science; sectors using data science; data science workflow; role of Python; NumPy, SciPy, Pandas, Matplotlib, Seaborn/Plotly; notebooks; data science project lifecycle.	12	15	0
II	Data Collection, Management, and EDA: Data collection through files, APIs, forms, scraping, and databases; data quality; knowledge check; exploratory data analysis; qualitative and quantitative EDA; graphical EDA; interpreting analytical conclusions.	11	15	0
III	Feature Generation and Basic Modeling: Feature generation and feature selection; extracting meaning from data; role of domain expertise; train-test split; introduction to predictive modeling; simple regression/classification demonstrations; evaluation basics.	11	15	0
IV	Visualization, Applications, Ethics, and Reporting: Visualization principles and tools; applications in business, healthcare, education, governance, and social media; data science and ethical issues; privacy and security; preparing a data science report and presentation.	11	15	0
	Total	**45**	**60**	**0**

Mode of In-Semester assessment:
A. Sessional Activity
B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Set up Python/Jupyter environment.
2. Perform NumPy and Pandas data manipulation.
3. Clean real-world CSV data and handle missing values.
4. Visualize data using Matplotlib/Seaborn/Plotly.
5. Create a mini-project involving EDA and simple prediction using scikit-learn.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain key data science concepts, workflows, tools, and applications.
CLO2. Collect, clean, and manage datasets for analysis.
CLO3. Perform EDA and visualization using Python libraries.
CLO4. Develop simple predictive models and discuss ethical issues in data science.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge		CLO1				
Procedural Knowledge			CLO2,CLO3			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	1	1	0	1	0	0	2	0
CLO2	2	2	0	0	0	0	0	0	1	0
CLO3	2	2	0	2	0	0	0	0	1	0
CLO4	2	2	3	0	2	0	0	0	1	0
Average Contribution	**2**	**2**	**1**	**0.75**	**0.5**	**0.25**	**0**	**0**	**1.25**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Skiena, S. S. (2017). The data science design manual. Springer.
2. VanderPlas, J. (2016). Python data science handbook. O'Reilly Media.
3. Trivedi, M. C. (2025). Data science and data analytics using Python. Khanna Publishing House.

Reference Books

1. Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media.
2. Wickham, H., & Grolemund, G. (2017). R for data science: Import, tidy, transform, visualize, and model data. O'Reilly Media.
3. Zumel, N., & Mount, J. (2014). Practical data science with R. Manning Publications.

Semester-V

DSE-5-BCA-502-D: Time Series Analysis

Course Code	**: DSE-5-BCA-502-D**
Course Title	**: Time Series Analysis**
Nature of the Course	**: DSE**
Stream	**: Data Science**
Total Credit assigned	**: 5**
Component of the Course	**: Lecture: 3 Practical: 2 Tutorial: 0**

Prerequisite : Probability, statistics, Python/R programming, and basic data visualization.

Course Objectives

The main objectives of this course are:

Objective-1. Understand time-indexed data, components, and forecasting needs.
Objective -2. Apply smoothing, decomposition, stationarity, autocorrelation, and classical forecasting models.
Objective -3. Evaluate forecasts and use basic machine-learning approaches for time-series prediction.

Course Content

Unit	Content	L	P	T
I	Introduction to Time Series Data: Time-indexed observations; cross-sectional vs. time-series data; trend, seasonality, cyclic and irregular components; plotting time-series; resampling, missing time stamps, lag features, moving averages, and rolling statistics.	12	15	0
II	Decomposition, Stationarity, and Correlation: Additive and multiplicative decomposition; seasonal adjustment; stationarity; differencing and transformations; autocorrelation and partial autocorrelation; white noise; diagnostics.	11	15	0
III	Forecasting Models: Naive and seasonal naive forecasts; moving average; exponential smoothing; Holt and Holt-Winters methods; AR, MA, ARMA, ARIMA, and SARIMA; model identification and residual checking.	11	15	0
IV	Time-Series Evaluation and Applications: Train-test split for temporal data; walk-forward validation; MAE, RMSE, MAPE; anomaly detection; introductory machine learning for time-series forecasting; applications in sales, finance, weather, education, and traffic analytics.	11	15	0
	Total	**45**	**60**	**0**

Mode of In-Semester assessment:
A. Sessional Activity
B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Visualize and decompose a time-series dataset.
2. Compute moving averages and exponential smoothing forecasts.
3. Check stationarity and plot ACF/PACF.
4. Fit ARIMA/SARIMA models and evaluate forecasts.
5. Build a simple machine-learning forecast using lag features.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Identify components and properties of time-series data.
CLO2. Apply decomposition, transformation, and stationarity techniques.
CLO3. Use classical models for forecasting time-indexed data.
CLO4. Evaluate and communicate forecast accuracy using suitable metrics.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge		CLO1				
Conceptual Knowledge					CLO4	
Procedural Knowledge			CLO2,CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	3	0	0	0	0	0	0	1	0
CLO2	2	3	0	0	0	0	0	0	0	0
CLO3	2	3	2	0	0	0	0	0	1	0
CLO4	2	3	0	0	0	1	0	0	0	0
Average Contribution	**2**	**3**	**0.5**	**0**	**0**	**0.25**	**0**	**0**	**0.5**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Nielsen, A. (2019). Practical time series analysis: Prediction with statistics and machine learning. O'Reilly Media.

2. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and practice (3rd ed.). OTexts.

3. Shumway, R. H., & Stoffer, D. S. (2017). Time series analysis and its applications: With R examples (4th ed.). Springer.

Reference Books

1. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: Forecasting and control (5th ed.). Wiley.

2. Brockwell, P. J., & Davis, R. A. (2016). Introduction to time series and forecasting (3rd ed.). Springer.

3. VanderPlas, J. (2016). Python data science handbook. O'Reilly Media.

Semester-V

Course Code	**:**DSE-5-BCA-503-D
Course Title	**:**Machine Learning
Nature of the Course	**:**DSE
Stream	**:**Data Science
Total Credit assigned	**:**5
Component of the Course	**:**Lecture: 3 Practical: 2 Tutorial: 0

Prerequisite: Python programming, linear algebra basics, probability, statistics, and data preprocessing.

Course Objectives

The main objectives of this course are:

Objective -1. Introduce supervised, unsupervised, and reinforcement learning.

Objective -2. Apply regression, classification, clustering, and dimensionality-reduction techniques.

Objective -3. Evaluate models, avoid overfitting, and interpret basic machine-learning results.

Course Content

Unit	Content	L	P	T
I	Foundations of Machine Learning: Learning paradigms; data representation; training, validation, and testing; bias-variance trade-off; overfitting and underfitting; feature scaling; evaluation workflow.	12	15	0
II	Supervised Learning: Regression and classification; linear regression; logistic regression; generalized linear models; decision tree; support vector machines; kernels; Naive Bayes; K-nearest neighbors; multiclass classification.	11	15	0
III	Unsupervised and Representation Learning: Clustering: K-means and hierarchical clustering; dimensionality reduction: PCA and kernel PCA; matrix factorization; anomaly detection; introduction to representation learning.	11	15	0
IV	Model Evaluation and Advanced Topics: Cross-validation; confusion matrix, precision, recall, F1-score, ROC-AUC; ensemble methods: bagging, boosting, random forest; transfer learning and reinforcement learning overview; introduction to deep learning and GPU use.	11	15	0
	Total	**45**	**60**	**0**

Mode of In-Semester assessment:

A. Sessional Activity

B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Preprocess a dataset for machine learning.
2. Implement regression and classification using scikit-learn.
3. Compare KNN, decision tree, SVM, and Naive Bayes models.
4. Apply clustering and PCA.
5. Build a mini-project with an evaluation report.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Select suitable ML algorithms for common problems.
CLO2. Train and evaluate supervised and unsupervised models.
CLO3. Apply dimensionality reduction and ensemble methods.
CLO4. Interpret model performance and limitations using appropriate metrics.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge				CLO1	CLO4	
Procedural Knowledge			CLO2, CLO3			
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	3	2	0	0	0	0	0	0	1	0
CLO2	2	2	2	0	0	0	0	0	2	0
CLO3	2	2	0	0	0	0	0	0	1	0
CLO4	2	2	2	0	0	0	0	0	1	0
Average Contribution	**2.25**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**1.25**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Murphy, K. P. (2012). Machine learning: A probabilistic perspective. MIT Press.
2. Jose, J. (2020). Introduction to machine learning. Khanna Book Publishing Company.
3. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.

Reference Books

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.
2. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
3. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning (2nd ed.). MIT Press.

Semester-VI

DSE-5-BCA-601-D: Big Data Analytics

Course Code : DSE-5-BCA-601-D
Course Title : Big Data Analytics
Nature of the Course : DSE
Stream : Data Science
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Prerequisite : Database management systems, Python/Java basics, and introductory data analytics.

Course Objectives

The main objectives of this course are:

Objective-1. Understand characteristics and challenges of big data.
Objective -2. Use distributed storage and processing concepts such as Hadoop, MapReduce, Spark, and NoSQL.
Objective -3. Design basic analytical workflows for large datasets.

Course Content

Unit	Content	L	P	T
I	Big Data Fundamentals: Big data characteristics: volume, velocity, variety, veracity, value; structured, semi-structured, and unstructured data; limitations of traditional relational systems; big data applications in web search, e-commerce, social networking, and education.	12	15	0
II	Distributed Storage and Processing: Distributed file systems; Hadoop ecosystem; HDFS; MapReduce programming model; dataflow and vertex-centric computing; fault tolerance; resource management; introduction to Hive and Pig-like query abstractions.	11	15	0
III	NoSQL and Spark Ecosystem: NoSQL models: key-value, column-family, document, graph; CAP theorem overview; Spark architecture; RDD/DataFrame concepts; Spark SQL; basics of streaming analytics; introduction to Storm/Giraph-like systems.	11	15	0
IV	Big Data Analytics Applications: Large-scale descriptive and predictive analytics; machine-learning pipelines on large datasets; recommendation and log analysis; visualization of big data; governance, security, privacy, and ethical issues in big data projects.	11	15	0
	Total	**45**	**60**	**0**

Mode of In-Semester assessment:

A. Sessional Activity
B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Install/use a local Hadoop or Spark environment.
2. Perform word count or log analysis using MapReduce/Spark.
3. Store and query data using a NoSQL database.
4. Perform Spark DataFrame operations on a large CSV dataset.
5. Create a mini-project report on big data analytics.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Explain big data architectures, characteristics, and challenges.
CLO2. Use basic distributed storage and processing methods.
CLO3. Compare relational and NoSQL approaches for large datasets.
CLO4. Build introductory analytics workflows for large datasets.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge		CLO1		CLO3		
Procedural Knowledge			CLO2			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	2	0	0	0	1	0	0	2	0
CLO2	2	2	0	0	0	0	0	0	2	0
CLO3	2	2	0	2	0	0	0	0	2	0
CLO4	2	2	3	0	0	0	0	0	2	0
Average Contribution	**2**	**2**	**0.75**	**0.5**	**0**	**0.25**	**0**	**0**	**2**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Jain, V. K. (2022). Big data and Hadoop. Khanna Book Publishing Company.

2. White, T. (2012). Hadoop: The definitive guide (3rd ed.). O'Reilly Media.

3. Minelli, M., Chambers, M., & Dhiraj, A. (2013). Big data, big analytics: Emerging business intelligence and analytic trends for today's businesses. Wiley.

Reference Books

1. Sadalage, P. J., & Fowler, M. (2012). NoSQL distilled: A brief guide to the emerging world of polyglot persistence. Addison-Wesley.

2. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). Mining of massive datasets (3rd ed.). Cambridge University Press.

3. Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). Learning Spark. O'Reilly Media.

Semester-VI

DSE-5-BCA-602-D: Exploratory Data Analysis

Course Code : DSE-5-BCA-602-D
Course Title : Exploratory Data Analysis
Nature of the Course : DSE
Stream : Data Science
Total Credit assigned : 5
Component of the Course : Lecture: 3 Practical: 2 Tutorial: 0

Prerequisite: Python/R, spreadsheet analytics, basic statistics, and data visualization.

Course Objectives

The main objectives of this course are:

Objective-1. Understand EDA as a systematic process for discovering structure in data.
Objective-2. Perform data profiling, cleaning, transformation, and feature engineering.
Objective-3. Use statistical summaries and visualizations to develop hypotheses and communicate findings.

Course Content

Unit	Content	L	P	T
I	EDA Concepts and Workflow: Purpose of EDA; data understanding; data dictionary; data types; data quality dimensions; exploratory vs. confirmatory analysis; reproducible notebooks and EDA reports.	12	15	0
II	Data Profiling and Preprocessing: Missing values, duplicates, outliers, inconsistent formats, transformations, encoding categorical variables, scaling, normalization, date-time feature extraction, and text data basics.	11	15	0
III	Statistical and Visual EDA: Univariate, bivariate, and multivariate analysis; measures of central tendency and dispersion; skewness and kurtosis; correlation and association; box plots, histograms, density plots, scatter plots, pair plots, heatmaps, and faceting.	11	15	0
IV	Feature Engineering and Communication: Feature generation and feature selection; dimensionality reduction with PCA; hypothesis generation; identifying patterns and anomalies; EDA for model readiness; communicating insights in reports and dashboards.	11	15	0
	Total	**45**	**60**	**0**

Mode of In-Semester assessment:
A. Sessional Activity
B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.
- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Create a data profiling report.
2. Clean and transform a messy real-world dataset.
3. Perform univariate, bivariate, and multivariate EDA.
4. Prepare feature-engineered data for machine learning.
5. Submit an EDA project report with visual narrative.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Profile and assess data quality for exploratory analysis.
CLO2. Clean and transform raw datasets using appropriate techniques.
CLO3. Use statistical and visual techniques for insight discovery.
CLO4. Prepare EDA outputs for modeling and decision making.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge				CLO1, CLO3		
Procedural Knowledge			CLO2			CLO4
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	0	2	0	0	0	0	0	0	1	0
CLO2	0	2	0	0	0	0	0	0	1	0
CLO3	0	3	0	0	0	0	0	0	1	0
CLO4	2	2	2	0	0	0	0	0	1	0
Average Contribution	**0.5**	**2.25**	**0.5**	**0**	**0**	**0**	**0**	**0**	**1**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Tukey, J. W. (1977). Exploratory data analysis. Addison-Wesley.

2. McKinney, W. (2022). Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media.

3. Wickham, H., & Grolemund, G. (2017). R for data science: Import, tidy, transform, visualize, and model data. O'Reilly Media.

Reference Books

1. Zheng, A., & Casari, A. (2018). Feature engineering for machine learning: Principles and techniques for data scientists. O'Reilly Media.

2. Galli, S. (2020). Python feature engineering cookbook. Packt Publishing.

3. VanderPlas, J. (2016). Python data science handbook. O'Reilly Media.

Semester-III

DSE-3-BCA-301-F: Web Programming -

Course Code : DSE-3-BCA-301-F
Course Title : Web Programming - I
Nature of the Course : DSE
Stream : Full Stack Development
Total Credit assigned : 3
Component of the Course : Lecture: 1 Practical: 2 Tutorial: 0

Prerequisite: Basic programming knowledge and familiarity with using a computer and Internet browser.

Course Objectives

The main objectives of this course are:

Objective-1. Understand web architecture, HTML, CSS, JavaScript, and responsive design.
Objective-2. Develop static and interactive client-side web pages.
Objective-3. Use DOM, events, validation, AJAX, XML/JSON, and basic deployment practices.

Course Content

Unit	Content	L	P	T
I	Web Fundamentals and HTML5: World Wide Web, HTTP/HTTPS overview, browsers, DNS, hosting, URLs; HTML document structure; text, links, images, lists, tables, forms, semantic tags, media, iframe, accessibility basics.	4	15	0
II	CSS3, Layout, and Responsive Design: CSS syntax, selectors, colors, typography, box model, positioning, display, flexbox, grid basics; internal/external/inline CSS; responsive design; Bootstrap or Tailwind basics; mobile-first design.	4	15	0
III	JavaScript Foundations: Variables, data types, strings, numbers, booleans, objects, arrays, operators, conditions, loops, functions, events, scope, error handling basics, date and string handling, browser console and debugging.	4	15	0
IV	DOM, AJAX, JSON, XML, and Client-Side Applications: DOM traversal and manipulation; form validation; regular expressions; AJAX/fetch; JSON keys, values, arrays, and objects; XML, DTD/schema and XSLT overview; integrating HTML, CSS, and JavaScript into small web applications.	3	15	0
	Total	**15**	**60**	**0**

Mode of In-Semester assessment:
A. Sessional Activity
B. Sessional Tests / Assignments

Sessional activity:

- Problem-based assignments related to the unit-wise course content.

- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Create resume, timetable, and college website pages using HTML.
2. Style pages using inline, internal, and external CSS.
3. Develop responsive pages using Bootstrap/Tailwind.
4. Build a login form with JavaScript validations.
5. Use AJAX/fetch to load data and render JSON/XML content.
6. Deploy a simple personal website.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Create structured HTML pages and forms.
CLO2. Apply CSS and responsive design techniques.
CLO3. Develop basic interactive web pages with JavaScript.
CLO4. Use JSON/AJAX/XML concepts in client-side web applications.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge		CLO4				
Procedural Knowledge			CLO2,CLO4			CLO1, CLO3
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	3	2	0	0	0	0	1	0
CLO2	2	3	3	3	0	0	0	0	1	0
CLO3	2	0	3	3	0	0	0	0	1	0
CLO4	2	0	2	3	0	0	0	0	1	0
Average Contribution	**2**	**0.75**	**2.75**	**2.75**	**0**	**0**	**0**	**0**	**1**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Lemay, L. (2016). Mastering HTML, CSS & JavaScript web publishing. BPB Publications.
2. Powell, T. A. (2017). HTML & CSS: The complete reference (5th ed.). McGraw-Hill Education.
3. Duckett, J. (2011). HTML and CSS: Design and build websites. Wiley.

Reference Books

1. Haverbeke, M. (2018). Eloquent JavaScript: A modern introduction to programming (3rd ed.). No Starch Press.
2. Moreto, S. (2016). Bootstrap 4 by example. Packt Publishing.
3. Alam, T. (2011). Web technologies. Khanna Book Publishing Company.

Semester-IV

DSE-3-BCA-401-F: Web Programming - II

Course Code : DSE-3-BCA-401-F
Course Title : Web Programming - II
Nature of the Course : DSE
Stream : Full Stack Development
Total Credit assigned : 3
Component of the Course : Lecture: 1 Practical: 2 Tutorial: 0

Prerequisite: Web Programming - I or equivalent knowledge of HTML, CSS, JavaScript, and basic programming.

Course Objectives

The main objectives of this course are:

Objective-1. Python/Django, or Ruby on Rails.
Objective-2. Use databases, REST APIs, authentication, sessions, and MVC patterns.
Objective-3. Build, test, deploy, and secure full-stack web applications.

Course Content

Unit	Content	L	P	T
I	Server-Side Programming and MVC: Client-server architecture; request-response lifecycle; routing; controllers; templates; MVC pattern; server-side forms; validation; sessions and cookies; selected framework setup.	4	15	0
II	Databases and RESTful APIs: Relational database basics; CRUD operations; ORM/ActiveRecord-like concepts; schema design; connecting applications to databases; REST architecture; API endpoints; JSON request/response; API testing.	4	15	0
III	Front-End Integration and Modern Frameworks: Asynchronous requests; consuming APIs; component-based front-end overview; React/TypeScript or comparable framework; state management basics; form handling; build tools and package management.	4	15	0
IV	Security, Testing, Deployment, and Production Readiness: Authentication and authorization; password hashing; CSRF/XSS/SQL injection prevention; unit/integration testing basics; Git workflow; environment variables; Docker/container overview; CI/CD basics; cloud deployment; logging, localization, and error monitoring.	3	15	0
	Total	**15**	**60**	**0**

Mode of In-Semester assessment:
A. Sessional Activity
B. Sessional Tests / Assignments
Sessional activity:

- Problem-based assignments related to the unit-wise course content.

- Short quizzes, practical demonstrations, and viva voce.
- Mini-project/report based on a dataset, software tool, or application area.

Evaluation Criteria:

- Correctness, completeness, and clarity of implementation.
- Appropriate use of tools, methods, and documentation.
- Ability to interpret results and communicate findings.

Laboratory Work:

1. Build a Todo or student-management application with CRUD.
2. Connect a web app to a relational database.
3. Create REST APIs and test them using Postman/Thunder Client.
4. Add authentication, sessions, and authorization roles.
5. Develop a front-end interface that consumes APIs.
6. Deploy a full-stack mini-project and submit documentation.

Expected Course Learning Outcomes:

By the end of the course, students will be able to:

CLO1. Develop server-side applications using an MVC framework.
CLO2. Design database-backed CRUD applications and REST APIs.
CLO3. Integrate front-end and back-end components.
CLO4. Apply basic security, testing, deployment, and production-readiness practices.

Cognitive Map of Course Learning Outcomes based on Revised Bloom's Taxonomy

Knowledge Dimension	**Remember**	**Understand**	**Apply**	**Analyse**	**Evaluate**	**Create**
Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge			CLO4			CLO1,CLO2,CLO3
Metacognitive Knowledge						

Mapping of PSO with CLO

CLO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9	PSO10
CLO1	2	0	3	3	0	0	0	0	1	0
CLO2	3	3	3	2	0	0	0	0	1	0
CLO3	2	0	2	2	0	0	0	0	1	0
CLO4	2	0	2	3	2	0	0	0	1	0
Average Contribution	**2.25**	**0.75**	**2.5**	**2.5**	**0.5**	**0**	**0**	**0**	**1**	**0**

(3=High, 2=Moderate, 1=Low, -=No correlation)

Suggested readings:

Text Books

1. Brown, E. (2019). Web development with Node and Express: Leveraging the JavaScript stack (2nd ed.). O'Reilly Media.

2. Flanagan, D. (2020). JavaScript: The definitive guide (7th ed.). O'Reilly Media.

3. Banks, A., & Porcello, E. (2020). Learning React: Modern patterns for developing React apps (2nd ed.). O'Reilly Media.

Reference Books

1. Holmes, S., & Harber, C. (2019). Getting MEAN with Mongo, Express, Angular, and Node (2nd ed.). Manning Publications.

2. Hartl, M. (2022). Ruby on Rails tutorial: Learn web development with Rails (7th ed.). Addison-Wesley Professional.

3. Freeman, E., & Robson, E. (2014). Head First JavaScript programming. O'Reilly Media.